TABLE OF CONTENTS

Contributed Articles

http://www.kdd.org/explorations/

Association for Computing Machinery
Advancing Computing as a Science & Profession

Blockchain for Large Language Model Security and Safety: A Holistic Survey

Caleb Geren[†][*], Amanda Board[‡], Gaby G. Dagher[♮], Tim Andersen[♮], and Jun Zhuang[♮]
[†]Lehigh University, Bethlehem, PA, USA, [‡]University of Idaho, Moscow, ID, USA
[♮]Boise State University, Boise, ID, USA

[†]cdg225@lehigh.edu,[‡]boar9227@vandals.uidaho.edu,
[♮]{gabydagher, tandersen, junzhuang}@boisestate.edu

ABSTRACT

With the growing development and deployment of large language models (LLMs) in both industrial and academic fields, their security and safety concerns have become increasingly critical. However, recent studies indicate that LLMs face numerous vulnerabilities, including data poisoning, prompt injections, and unauthorized data exposure, which conventional methods have struggled to address fully. In parallel, blockchain technology, known for its data immutability and decentralized structure, offers a promising foundation for safeguarding LLMs. In this survey, we aim to comprehensively assess how to leverage blockchain technology to enhance LLMs' security and safety. Besides, we propose a new taxonomy of blockchain for large language models (BC4LLMs) to systematically categorize related works in this emerging field. Our analysis includes novel frameworks and definitions to delineate security and safety in the context of BC4LLMs, highlighting potential research directions and challenges at this intersection. Through this study, we aim to stimulate targeted advancements in blockchain-integrated LLM security.

1. INTRODUCTION

The widespread application of large language models (LLMs) has progressed at an unprecedented pace and scale in our daily lives [111]. Such a widespread application exposes several vulnerabilities inherent to LLMs, such as data poisoning [55; 36], prompt injections [77; 101], and hallucinations [53; 136; 86; 12]. For example, prompt injections can exploit a model's propensity to disclose information, resulting in significant data leakage, like leaking personally identifiable information (PII) to an unauthorized user [123; 20]. Although numerous studies have attempted to address these issues [129; 2], there remains no effective mitigation strategies capable of addressing growing concerns about these issues in LLMs [123; 39; 71]. Typically, defensive strategies against these threats are implemented through established machine-learning methods, such as applying differential privacy (DP) techniques to the entire dataset to enhance privacy protections [1; 127]. While DP strategies are one of the crucial applications, these strategies cannot fully guarantee data privacy in LLMs [71] due to DP's ability to protect primarily "by whom" data is contributed, rather than "about whom" the data is focused on. Additionally, another common attempt to tackle the data privacy problem in LLMs is to apply federated learning (FL) techniques in the training process by distributing model training across multiple nodes to create a decentralized environment [78]. Naturally, this lends itself to further obscuring sensitive information in a model's corpus. However, it has been shown that by taking model weights or gradients, original data from the model can still be reconstructed [65]. What's worse, federated-learning approaches are susceptible to many of the same types of attacks as large language models, such as single-point-of-failure attacks or man-in-the-middle attacks [89]. This trend of typical approaches failing to exhaustively defend against the range of attacks now affecting LLMs continues across multiple traditional threat/defense models [99; 139; 127].

To address the limitations of the above-mentioned methods and further enhance data privacy, blockchain technology has emerged as a promising solution [80]. It ensures data integrity through various tamper-evident mechanisms, introduces a high level of confidentiality to otherwise centralized systems, and guarantees data provenance by enabling traceable and auditable records [27; 19; 4]. These benefits can significantly strengthen the robustness of large language models. Integrating blockchain technology lays the foundation for stronger privacy protection, enhanced inference validation, defenses against adversarial attacks, and other security measures to be incorporated into the design of large language models. This overlapping research direction is still in its early stages. To facilitate a deeper understanding of the current landscape for emerging researchers, we conduct a comprehensive literature review in this paper to explore how blockchain technology can better serve large language models (BC4LLMs). Overall, our objectives in this survey paper are to address the following four research questions:

RQ1. What are the pressing LLM-related security concerns that may be addressed with blockchain technology?

RQ2. How can we meaningfully differentiate between security and safety in the context of BC4LLMs?

RQ3. In what ways can blockchain technology be used to enhance the safety of LLMs?

RQ4. What are prominent gaps within the BC4LLMs area, how can these gaps influence research directions, and what resources can we provide to enable potential new directions?

[*] First two authors contributed equally to this work.

Table 1: **Overview of Existing Related Surveys.** We compare related surveys about blockchain techniques and LLMs from various perspectives, such as background, threat model, definition, security, safety, etc. In particular, we are interested in investigating whether (i) relevant subjects are discussed in the background section, (ii) a model of threat categorization is introduced, (iii) definitions of security and safety are proposed, (iv) security and/or safety with regards to BC4LLMs is explored, (v) future BC4LLMs work is probed, (vi) the survey focuses on LLMs for Blockchain. We denote ●, ◐, and ○ as a full, partial, and no discussion of the corresponding items.

Source	*LLMs and BC Background*	*Threat Model*	*Definitions*	*Security in BC4LLMs*			*Safety in BC4LLMs*	*Future Work*	*LLMs for Blockchain*
				LLM	*AI*	*Non-AI*			
Luo, et al. 2023 [72]	●	◐	○	◐	◐	◐	◐	●	○
Mboma, et al. 2023 [76]	●	○	○	◐	○	◐	◐	○	●
He, et al. 2024 [41]	●	○	○	○	○	○	○	○	●
Heston 2024 [43]	◐	○	○	●	◐	○	◐	◐	◐
Salah, et al. 2019 [94]	◐	○	○	○	●	●	◐	○	○
Bhumichai, et al. 2024 [13]	◐	○	○	○	●	◐	◐	○	○
Dinh and Thai 2018 [28]	◐	○	○	○	●	○	◐	○	◐

Notably, we focus specifically on how blockchain systems may impact large language models' *security* and *safety*. By narrowing our scope to these dimensions, we aim to provide a more detailed analysis and categorization of seemingly disparate works, thereby encouraging targeted research advances in specific directions. In general, attacks on LLMs typically manifest in two primary ways: as direct exploitations by malicious third parties that capitalize on system vulnerabilities (security) [5; 126; 139; 44; 68] and as inherent risks embedded within LLM structures that expose users to potential harm without external malicious influence (safety) [33; 114; 95; 130]. We base our analysis of the blockchain for LLMs (BC4LLMs) on this critical distinction between security and safety, a distinction that we underscore through explicit definitions contextualized for LLMs. To the best of our knowledge, this is the first study to rigorously define these terms in the BC4LLMs context, providing a foundation for subsequent work in this area. Furthermore, we contribute to the discourse on privacy in LLMs by delineating active and passive privacy efforts, modeled after a survey about data privacy [127].

To distinguish our analysis of BC4LLMs from other similar works through the lenses of safety and security, **we compare several reviews** in this domain. He et al. [41] examine the relationship between LLMs and blockchain in analyzing how LLMs can further enhance blockchain systems. Mboma et al. [77] provide an exploratory review of general integrations between blockchain and large language models, which is similar to Heston's analysis of integrating the two technologies in telemedicine [43]. Additionally, Salah et al. [94], Bhumichai et al. [13], and Dinh et al. [28] provide an overview of potential and existing technologies between blockchain and artificial intelligence in general. In short, current reviews that specifically address blockchain and LLMs lack a clear focus on the specific applications of these technologies, whereas broader reviews that encompass blockchain and AI sacrifice the depth of analysis. To close this gap, we present an overview of related surveys in Table 1, which juxtaposes the above papers' contents with our specific focuses, highlighting the distinction in our study.

We outline our main **contributions** and highlight the impact to answer our research questions as follows:

1. In this work, we first contribute a series of frameworks, definitions, and compiled resources. Most prominently, we propose a new taxonomy about applying blockchain techniques for LLMs in Figure 3. Through the proposed taxonomy, we aim to succinctly explain the relevant interactions between the blockchain techniques and corresponding LLMs' vulnerabilities [**RQ1**][**RQ3**]. To further contextualize this taxonomy and ground our discussion of existing literature, we propose two foundational definitions of safety and security specific to LLMs [**RQ2**]. Moreover, we provide a collection of datasets relevant to BC4LLMs, equipping future researchers with resources to build on the connections delineated by our taxonomy and informed by our definitions [**RQ4**].

2. We also highlight additional components of our paper that, while supporting our main contributions, serve as valuable artifacts in the BC4LLMs space. One such artifact is our definition of specific areas within the broader concept of safety, further detailed in Table 3 [**RQ2**][**R3**]. These definitions reinforce our conceptualization of safety for LLMs. To enhance our definitions of both safety and security, we specifically address privacy within the security context, reaffirming two terms introduced by Yan et al. [127]: passive and active privacy [**RQ1**][**RQ3**]. Besides, our contextualization of LLMs within various AI sub-fields [**RQ1**] and our concise taxonomic overview of blockchain components [**RQ1**][**RQ3**] hold intrinsic value as distinct contributions to the field.

3. Last, we conduct a comprehensive literature review in Section 4, where we classify research works across several interrelated domains, offer novel insights within these categorized domains, and align all BC4LLMs research projects with LLMs' safety and security. By conducting this review, we provide an informative perspective on the potential of utilizing blockchain techniques to enhance LLMs [**RQ1**][**RQ2**][**RQ3**][**RQ4**].

The remaining sections are organized as follows: In Section 2, we introduce the background of blockchain technology and large language models. In Section 3, we describe our methodology, including the criteria used to filter works for this review and the relevant definitions that guide our analysis of the current literature. We also share our model of threat categorization, which aligns with the categorization proposed by Yao et al. [129]. In Section 4, we conduct a comprehensive literature review of BC4LLMs in safety and security, examining key works in relation to our proposed taxonomy. In Section 5, we present datasets relevant to BC4LLMs. In Section 6, we address key challenges at the intersection of blockchain technology and LLMs that hinder advancement in this area. In Section 7, we discuss future research directions within the field of BC4LLMs. Finally, in Section 8, we summarize our efforts, providing a holistic view of the current progress of BC4LLMs.

2. BACKGROUND

In this section, we present an overview of blockchain as a distributed ledger technology and relate the abilities of large language models to their capacities as agents with respect to their nature as both AI models and their tendency to interact with vast quantities of data.

2.1 Blockchain

Since Satoshi Nakamoto [82] introduced Bitcoin as a decentralized currency in 2008, there has been a subsequent explosion of academic and commercial interest in its underlying blockchain technology [59; 105; 104]. Additionally, and as highlighted by the introduction of Vitalik Buterin's Ethereum blockchain in 2014 [17], there has been a particular focus on blockchain's potential applications in fields entirely disparate from digital currencies. The interest in blockchain, or distributed ledger technologies, stems from its guarantees of data sovereignty, transparency, and relative permanence. Concisely, these properties are often referred to as immutability and irrefutability. Ranging from many diverse fields such as health care record management, digital identity management, or tax auditing, these properties are widely applicable and highly desirable, even though the mechanisms through which we achieve them can be somewhat complex and opaque. In light of the oftentimes convoluted nature of blockchain systems, we introduce blockchain to the reader in a piecemeal fashion in order to emphasize the modular, yet interconnected nature of such systems. Figure 1 represents an overview of our characterization of blockchain systems in general. We purposefully exclude certain components such as the incentive mechanism, or wallets, as they are beyond the scope of our analysis of blockchain as a means to serve large language models.

2.1.1 Blockchain Components

Consensus Protocol. Of particular interest to BC4LLMs, and arguably the most fundamental component within a blockchain, the consensus protocol is the governing system that controls how data is added to a blockchain's ledger. At its core is the consensus mechanism, which both ensures the validity of proposed data and fosters an environment of accountability, so that nodes submitting invalid information may be penalized accordingly. For example, the Proof of Work (PoW) consensus mechanism [82] is by far the most widely known. In it, nodes must solve a complex mathematical equation in order to gain rights to propose data for the blockchain. When such a node submits new data, it is scrutinized by every other node in the system. If the data is malicious, or untruthful, the proposal is rejected and the corresponding processing power performed by the malicious node has effectively been wasted, as that node will not receive the incentive, a Bitcoin reward. The underlying ideas of accountability, certain nodes being selected as 'block proposers', and the 'proof' of the ability to submit information to the chain are central ideas in consensus protocols across blockchains with different consensus protocols [84].

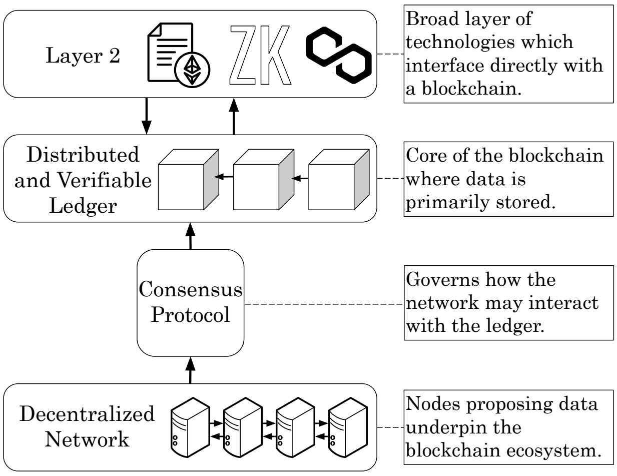

Layer 2
ZK
Broad layer of technologies which interface directly with a blockchain.
Distributed and Verifiable Ledger
Core of the blockchain where data is primarily stored.
Consensus Protocol
Governs how the network may interact with the ledger.
Decentralized Network
Nodes proposing data underpin the blockchain ecosystem.

Figure 1: A blockchain consists of four main components. A decentralized network of nodes interacts with a ledger via a governing consensus mechanism. This ledger, adequately protected by the consensus mechanism, creates what we refer to as the blockchain. Layer 2 solutions can interface with this ledger to enable greater functionality between users and a blockchain's data.

Verifiable Ledger. At a blockchain's core sits the verifiable ledger, a repository of data bolstered by a secure way of maintaining the integrity of that data. Of note is the particular technique through which data itself is verified on the ledger: the Merkle tree [79], or a variation thereof. Typically implemented as a ground-up binary tree, data is stored in leaf nodes, with hashed pointers of that data cascading up the tree. This structure results in a comprehensive 'Merkle root', a hash pointer consisting of all the other hash pointers in lower levels of the tree, which is ultimately based on the data stored in the leaf nodes. This technique ensures the integrity of information in the leaf nodes, as any alteration to the data is instantly reflected in the Merkle root. Likewise, new additions to the Merkle tree can be checked against previous states of the tree via a recalculation of the Merkle root accounting for the new transactions. This technique, complementing the verifiable ledger, is often the key to LLM data provenance and traceability solutions that rely on blockchain technology.

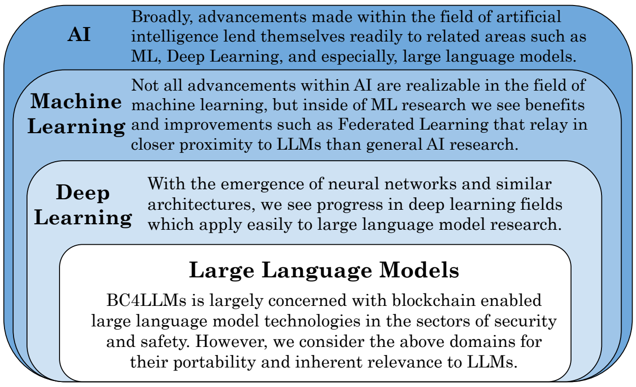
Decentralized Network. Critically, blockchains are decentralized networks. That is, no central server or group of servers may assume control of the network in a way that would compromise the network's state of trustlessness. This is achieved through multiple avenues, such as the aforementioned consensus protocol, the distribution of the verifiable ledger among a large number of independent nodes, and the accessibility of a given blockchain's network. [27] In this way, no users in the network are required to trust any other user. This fundamental aspect of blockchain is responsible for already realized and potential advancements with LLMs concerning areas such as RAG, the training process, and even supply chain issues.

Layer 2 Technologies. Apart from the fundamental components found within all blockchains, several external architectures interface with blockchains and further enhance their applicability. Typically, these external architectures are referred to as layer 2 technologies, as they sit a 'layer' above the 'layer 1' blockchain. Increasingly relevant as blockchain's influence grows, layer 2 solutions are a burgeoning area with numerous novel research directions. Most prominent among these are smart contracts, scripts that rely on a blockchain's security guarantees to facilitate off-chain transactions [146]. Also, in layer 2, zero-knowledge rollups are often combined with the efficacy of smart contracts. Often used to strengthen scalability, zero-knowledge rollups batch unproposed transactions together, and instead of submitting the transactions themselves, submit proof that the transactions are indeed valid [108]. This allows for transactions to be added on-chain without the need for every full node to redo the calculations found within those transactions. This area of layer 2 technologies is pivotal as it relates to BC4LLMs - layer 2 has the necessary dynamism to react quickly to new and emerging LLM vulnerabilities.

2.2 Large Language Models

In recent years, large language models (LLMs) have emerged as a pivotal force in artificial intelligence (AI), contributing to widespread applications across diverse fields, such as trustworthiness [25; 51; 52; 67], scholarly document processing [148], signal processing [91], quantum computing [60], climate production [62; 61], software engineering [145], and healthcare [41] among multiple other learning environments. Zhao et al. [141] and Yang et al. [128] define LLMs and pre-trained language models (PLMs) from the perspectives of model size and training approach. Generally speaking, PLMs refer to language models that are pre-trained on large amounts of general text data and then fine-tuned for specific tasks. LLMs are a kind of PLM. The key distinction is that LLMs are generally larger in scale with more parameters. These large language models have demonstrated the ability to learn universal representations of language, used in various natural language processing (NLP) tasks [47], bolstering their applicability.

We discuss connections and following developments between AI, machine learning (ML), deep learning (DL), and LLMs in Figure 2. AI refers to a broad technique that aims at simulating human intelligence, encompassing a variety of approaches and methods. Machine Learning (ML) is a subfield of AI that develops algorithms and statistical models to automatically learn from data and efficiently perform specific tasks without the use of explicit instructions [64]. Deep Learning (DL) is a subset of ML that utilizes multi-layered neural networks to learn latent representation on various tasks [96]. LLMs are one of the popular applications using cutting-edge DL models, advancing natural language understanding and generation at the human level. In the following subsections, we elaborate on the process of how LLMs are pre-trained and utilized as safe and powerful AI systems.



Broadly, advancements made within the field of artificial intelligence lend themselves readily to related areas such as ML, Deep Learning, and especially, large language models.

Not all advancements within AI are realizable in the field of machine learning, but inside of ML research we see benefits and improvements such as Federated Learning that relay in closer proximity to LLMs than general AI research.

With the emergence of neural networks and similar architectures, we see progress in deep learning fields which apply easily to large language model research.

BC4LLMs is largely concerned with blockchain enabled large language model technologies in the sectors of security and safety. However, we consider the above domains for their portability and inherent relevance to LLMs.

Figure 2: The connection among AI, Machine Learning (ML), Deep Learning (DL), and LLMs.

2.2.1 *Model Training*

During the pre-training phase, the LLM is trained on a diverse, large dataset of textual data from various sources to learn the statistical properties of language. The LLM is equipped with a myriad of adjustable parameters, commonly reaching more than ten billion [47]. Due to the huge model size and the vast amount of data used to train it, it is computationally challenging to successfully train a capable LLM, requiring distributed training algorithms for learning the model parameters [141]. Another crucial factor for LLM training is the data itself. Data that models are trained on come from a wide variety of sources, but the data itself may not be up to date [106]. To mitigate this shortcoming, recent advancements have introduced Retrieval Augmented Generation (RAG), which is designed to augment and rectify the information returned by LLMs by consulting up-to-date online sources. The data that the LLM was trained on also has other deficiencies, like knowledge gaps in healthcare fields where data is private and restricted [50]. Due to these knowledge gaps, the LLM may conjure up hallucinations where the model generates false information during prompting [75; 3] because of a lack of relevant information. However, hallucinations may also occur with a plethora of data available as they are inherent problems in LLMs. Methods of preventing these hallucinations are elaborated in Section *4.2.2*. RAG can help rectify hallucinations, and fill in the gaps of data the LLM is missing, by using up-to-date and validated information from trustworthy online resources. This data retrieval method introduces novel vulnerabilities since the information gathered by the retriever is largely unaudited and may contain poisoned data or data that can lead to unsafe responses from the LLMs.

2.2.2 *Model Tuning and Utilization*

After pre-training, the parameters of LLMs can be further updated by training on domain-specific datasets in downstream tasks. This process is known as fine-tuning (FT) [16]. A kind of fine-tuning method called supervised fine-tuning (SFT), aims to improve LLMs' responsiveness to instructions, ensuring more desirable reactions involving three major components of instructions, inputs, and outputs. Inputs relate to prompting and the inputs depend on the instruc-

Table 2: **Differences in Definitions of Safety.** There is no unifying definition of safety within the area of large language models. We see obvious agreement that models should be law-abiding, ethical, and non-violent in order to be safe, and as such these properties are strongly relevant to our definition of safety. However, beyond that point, there is generally a deviation between the authors' respective definitions. This creates two further categories of terms, properties that are moderately relevant to safety and those that are weakly relevant. Questions of fairness, the informing ability of an LLM, and robustness are generally covered but not unanimously, and hence are moderately relevant, whereas privacy-preserving properties or non-sycophancy are rarely discussed in the current literature and are thus weakly relevant to safety. This dialogue between different modes of thought concerning what makes a large language model "safe" heavily influences our definition of safety and our resulting discussion.

Relevance	**Property**	*Sun et al.* [107]	*Liu et al.* [69]	*Han et al.* [38]	*Röttger et al.* [92]	*Zhang et al.* [137]	*Wang et al.* [115]	*Tedeschi et al.* [110]	*Inan et al.* [49]	*Weidinger et al.* [117]
Strong	Ethical	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Law-abiding	✓	✓	✓	✓	✓		✓	✓	✓
	Non-violent	✓	✓		✓	✓	✓	✓	✓	✓
Moderate	Fair				✓	✓	✓	✓		✓
	Informing		✓			✓	✓			✓
	Robust				✓	✓	✓			✓
Weak	Privacy Preserving	✓			✓		✓			
	Non-sycophantic				✓			✓	✓	

tions, similar to applications of open-ended generation in ChatGPT. By providing both inputs and outputs they form an instance, and multiple instances can exist for a single instruction [41]. Among fine-tuning, other training techniques within model prompting include instruction tuning and alignment tuning. By FT from a mixture of multi-task datasets formatted via natural language descriptions with the use of instruction tuning, LLMs are enabled to follow task instructions for new tasks without needing explicit examples, highlighting the ability of generalization for instruction following [141]. However, LLMs can demonstrate versatility, even without FT where they produce a phenomenon known as zero-shot learning, exhibiting the ability to perform tasks for which the model was never explicitly trained [16].

Alignment tuning, equipped with reinforcement learning, is used to enhance LLMs to be safe interactive models. Since LLMs are trained to capture the data characteristics of uncurated pre-training corpora involving both high-quality and low-quality data, the LLM can generate toxic, biased, or harmful content for humans. To mitigate this problem, an FT process based on reinforcement learning from human feedback (RLHF) is used to align the LLM with the outcomes that satisfy human values [141]. The RLHF process ranks LLM outputs, with rewards scaled to positive and negative values. The LLM is then trained to produce highly-ranked responses and avoid low-ranked responses. In healthcare, RLHF provides advantages to the model such as improved accuracy and reliability through continuous feedback from medical professionals, and customizes the interactions based on real clinical settings and patient needs [40]. These advanced training techniques improve LLM's ability to generalize across tasks and improve their overall utility in various domains.

3. RESEARCH METHODOLOGY

The discussion of blockchain technology's incorporation into large language models necessitates a corresponding exploration into the implications of various terms and definitions found at that intersection. For example, due to the rapid emergence of LLMs, there exists an absence of consensus in describing common phenomena concerning LLM safety and security. To ameliorate this effect, we take care to stress opposing, but related, definitions of safety found within many different works in Table 2. In light of these distinctions, we offer two formal definitions of security and safety in order to contextualize these differing but similar areas of research. These definitions will also serve to highlight where particular blockchain technologies could be applied in their respective domains, and focus research efforts.

First, and to allow a richer discussion centered around safety and security, we delineate between active and passive privacy within LLMs as introduced in [127].

- *Active privacy* is where a user intentionally tries to gain access to sensitive information by breaking the large language model, especially with backdoor attacks, prompt injection attacks, and membership inference attacks during the pre-training and FT phases.
- *Passive privacy* is the state or condition of any impacted person being protected from accidental or unexpected data leakage originating from a large language model. This definition includes protecting the privacy of not only users but people whose information was added to a model's corpus without their knowledge or consent.

Next, we introduce our definitions of security and safety regarding LLMs.

Definition 1. ***LLM Security.*** *A large language model is considered secure if it:*

1. *Withstands applicable adversarial attacks and maintains system integrity, providing consistent and accurate responses, and*
2. *Ensures active user privacy, explicitly resisting backdoor, prompt injection, and inference attacks to prevent malicious users from extracting private information.*

Table 3: **Safety Area Definitions and Examples.** The area immediately surrounding BC4LLMs lacks a unifying definition of safety as well as consensus on what terms within that definition precisely mean. We provide generalized definitions for terms considered in our definition, as well as examples of incidents in literature where LLMs deviate from behavior as described in the definition. Italicized terms indicate inclusion in our definition of safety.

Safety Area	Definition	Example of Non-alignment	
Ethicality	LLMs aligning with moral principles.	A LLM agreeing with eugenics. [42].	Found within definition of safety
Legality	LLMs refusing to assist users in illegal endeavors.	A LLM assisting a user in creating incendiary devices. [112].	
Non-violence	LLMs soliciting generally non-violent advice or instructions.	A LLM advising a user to perform a 'raid on a drug house' and 'kill everyone there' [34].	
Passive Privacy	LLMs protecting private data within their corpus absent of malicious threats.	A LLM partially or fully reconstructing private images from a given dataset [20].	
Honesty	LLMs refraining from producing inaccurate or misinformed responses which may lead to negative outcomes.	LLMs administering faulty or fundamentally dangerous advice to patients or physicians in a healthcare setting [85].	
Fairness	LLMs ensuring a equitable environment for interaction, regardless of social identity.	LLMs associating "male" names with qualities of leadership, and "female" names with qualities of amicability [114].	
Robustness	The ability of the LLM to defend against adversarial attacks, originating from outside the model. This is a wide-reaching term, and falls within our discussion of security as it relates to LLMs.	A LLM falling victim to a backdoor attack planted in poisoned training data and producing malicious outputs as a result [129].	
Non-sycophancy	LLMs choosing consistent outputs despite the chance that they may be in conflict with a user's beliefs or desires.	A LLM revising a correct answer to an incorrect answer after the user asks the LLM if they are sure or challenges the LLM's result in some way [98].	

Definition 2. ***LLM Safety.*** *A large language model is considered safe if it interacts with users in a trustworthy manner, adhering to the aforementioned (Table 3 interrelated properties of safety: being ethical, law-abiding, nonviolent, fair, passively privacy-preserving, and informing.*

These definitions will serve a versatile role throughout this paper as building blocks for our contextualization of relevant and notable research efforts in BC4LLMs. Besides, they will serve the community at large in helping to establish reliable and tangible properties of secure and safe large language models. Furthermore, They will help establish tighter definitions of finer-grained terms and ideas within BC4LLMs. For example, in Table 2, we provide definitions for terms found within our definition of safety to lessen the effect of the vague nature of some of the words. These definitions are backed by relevant examples found in the literature.

3.1 Research Approach and Limitations

Literature surveys often are limited in their depth and scope by unconscious factors that impact the authors' ability to fairly select papers for review. To be transparent, and to aid researchers conducting similar or future reviews, we outline our research approach and its associated limitations. While conducting our research, we used the search engine, Google Scholar, and several databases, including ACM Computing Surveys, IEEE Xplore, SpringerLink, and arXiv. We chose these databases as they either produce quality research and contribute to the growth of interest in novel areas, or in the case of arXiv have the most up-to-date papers available. With Google Scholar, we used keyword searches such as "blockchain for LLMs" and "blockchain-based LLMs" as starting points for relevant, intriguing research papers. To solve problems concerning the scope and interrelated domains of disparate areas, we gathered various applications of blockchain for AI, blockchain-enabled machine learning, federated learning, and deep learning tactics to apply them to LLMs. Lastly, of note is the fact that we were largely aided in this further research effort by a waterfall approach to finding research papers. That is, we found several foundational papers in the BC4LLM field, explored citations in those papers, and subsequently explored citations in those secondary papers. We continued investigating relevant citations in this waterfall fashion until we reasonably exhausted all relevant articles. Admittedly, this method of finding prominent research articles is limited in its natural tendency to develop blind spots to less well-known research articles or venues. However, in the spirit of a literature survey, we choose to focus on more established papers that more accurately capture the trends currently found in the space. For our exclusion criteria, we limited our research as follows: no duplicates; found articles from 2016 and above, excluding the original Merkle Tree paper [79]; no Masters or Ph.D. theses; and only studies written in English.

3.2 Model of Threat Categorization

There exists a wide variety of threats that affect LLMs. Oftentimes, many of these threats originate from the nature of LLMs acting as AI systems. In Figure 3, we refer to these vulnerabilities similarly to that discussed in [129], which categorized the most LLM vulnerabilities and AI-inherent vulnerabilities together, yet also included external threats under non-AI inherent vulnerabilities. We contribute further by applying these vulnerabilities to each respective process within developing a large language model and tying these

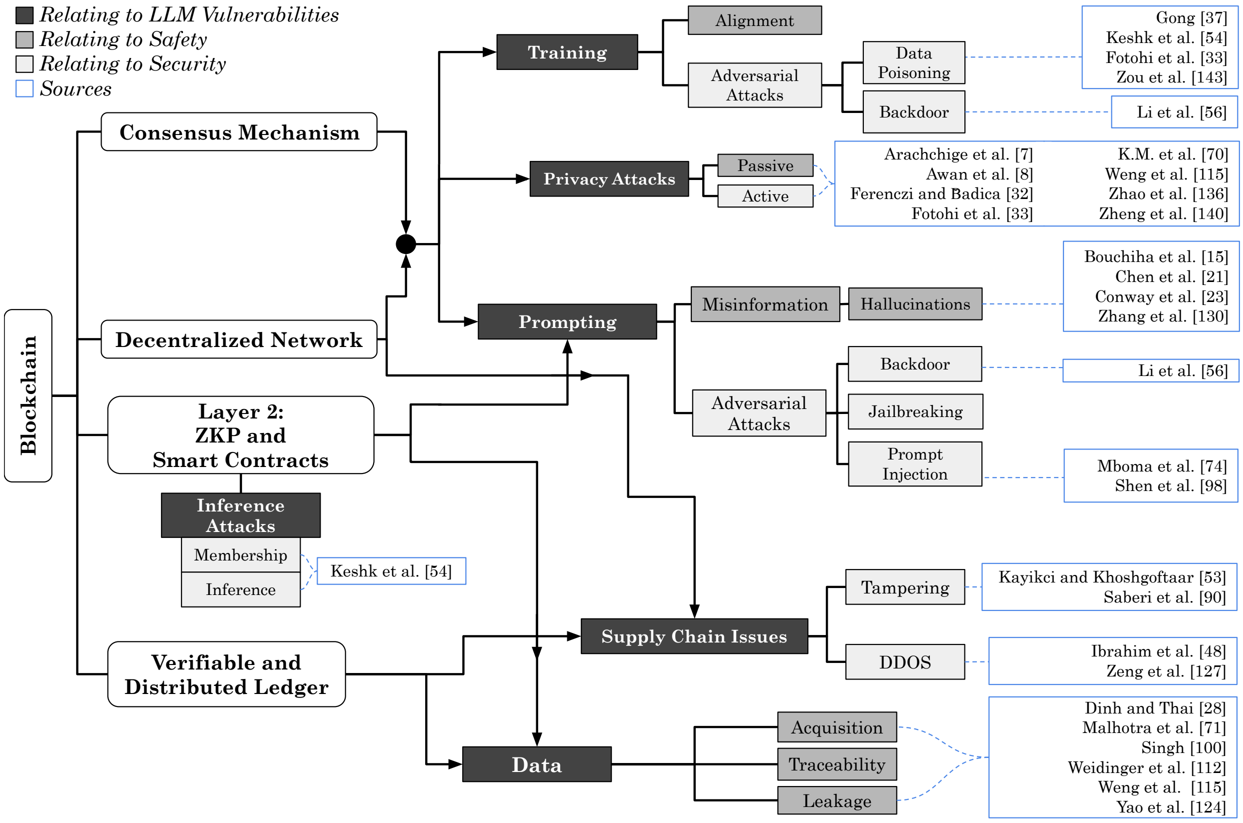


Figure 3: Taxonomy of Blockchain for LLM's Security and Safety. This diagram outlines the integration of blockchain technology to enhance the security and safety of large language models through categorizing interactions and safeguards into several layers and components. Each section supports relevant sources for further reference, illustrating a comprehensive approach to mitigating and preventing vulnerabilities as well as supplementing the security and safety of LLMs through blockchain technology. Promising areas that currently do not have blockchain as a solution to these vulnerabilities intentionally do not have source boxes.

to respective applications of blockchain. Beginning with the training process, LLMs are prone to threats such as data poisoning and backdoor attacks. As defined by Yao et al. [129], data poisoning is where attackers influence the training process by injecting malicious data into the training set, introducing vulnerabilities within the security and effectiveness of the model. Following the trend of poisoned data, there can be backdoor attacks implemented on the training data, as defined by Li et al. [57], who categorize backdoor attacks into attacks on training data and attacks on local models. The backdoor attacks on training data are further divided into attacks based on label flipping and attacks based on planting triggers. Attacks based on label flipping focus on manipulating the labels, whereas attacks on planting triggers modify the input data and labels, effectively constructing an adversarial sample. Then, attacks on local models are further divided into attacks based on modifications to the training process and attacks based on manipulating the trained model [57]. The backdoor attacks can be applicable to both the training and prompting phases of LLMs when using this distinction.

RAG attacks have a variety of issues, including privacy issues [132; 5] and knowledge poisoning attacks [149]. For RAG specifically, Xue et al. [126] propose BadRAG to identify security vulnerabilities, exposing direct attacks on the retrieval phase from semantic triggers, and uncovering indirect attacks on the generative phase of LLMs that were caused by a contaminated corpus. These RAG-specific attacks and defenses are elaborated on in Section 4.1.2. When interacting with an LLM, AI's inherent vulnerabilities become evident, as highlighted in [129], since LLMs are fundamentally AI models themselves. We focus on the prevalent adversarial attacks that malicious users may use to tamper with the LLM, attempt to find out sensitive information, or ruin the system entirely. We recognize jailbreaking and prompt injection as two separate but similar types of adversarial attacks that are initiated within prompting. For instance, jailbreaking prompts are designed to bypass the restrictions set by service providers during model alignment or other containment approaches [99]. Prompt injections aim to override an LLM's original prompt and direct it to follow a set of malicious instructions, leading to erroneous advice or unauthorized data leakage [68]. In Sections 4.2.1 and 4.2.2, we discuss instances of misinformation and passive privacy leakage addressed as safety concerns. Note that we include backdoor attacks based on modifications to the trained model in prompting since these backdoor attacks can still happen after model training [57].

Another relevant attack is a membership inference attack (MIA), a type of privacy attack where some malicious users, given access to the model, can determine whether a given point was used to train that model with high accuracy [83]. However, Neel and Chang [83] state that this attack is more related to information about the training point data leaking through the model, and that malicious users must have access to a candidate point in order to run the attack. Therefore, this attack is more prevalent with passive privacy, highlighting the need to prevent data leakage. Similar attacks are user inference attacks that seek to gain knowledge or insights about the model or data's characteristics, often by observing the model's responses or behavior [129].

Last but not least, we explore denial of service (DoS) attacks and supply chain vulnerabilities. Yao et al. [129] describe DoS attacks as a type of cyber attack that aims to exhaust computational resources, resulting in latency or making the technology resources unavailable. In this survey, we focus on distributed denial of service attacks (DDoS), a type of DoS attack where requests flood the system, attacking simultaneously from multiple sources on the network [29]. Yao et al. [129] also defined LLM supply chain vulnerabilities as the risks in the lifecycle of LLM applications that may occur from using vulnerable components or services, including third-party plugins that may be used to steal chat histories, access private information, and or execute code on a user's machine. All of these security vulnerabilities are substantial threats to LLMs that need to be mitigated or prevented. Possible methods of defense are discussed in Section 4.1, using current blockchain frameworks and experiments for these security problems, as listed by each developmental phase of the LLM, AI inherent threats, and supply chain issues.

4. EXISTING LITERATURE ON BC4LLM

Independently, the fields of both LLMs and blockchain research have grown substantially over the past several years. It is no surprise that the literature surrounding these topics has begun to morph and relate to each other. In previous research, we have seen LLMs for Blockchain Security [41] as well as an introduction to the term BC4LLM in Luo et al [72] where they provide a comprehensive survey of blockchain for LLMs. However, they do not acknowledge the multitude of safety and security solutions that blockchain provides for certain LLM vulnerabilities. Effectively, Luo et al. [72] aim to introduce BC4LLM for trusted AI, enabling reliable learning corpora, secure training processes, and identifiable generated content. In juxtaposition, our survey aims to analyze possible BC4LLM solutions closely related to our definitions of safety (2) and security (1) when looking at inherent system vulnerabilities in LLMs. To begin our analysis, we define these security problems based on previous work and highlight areas of research that are applicable to areas of BC4LLM safety and security.

4.1 Blockchain for LLMs' Security

Few papers and experiments analyze how the integration of these two technological powerhouses interacts with one another. We have seen benefits of this integration that apply to our definition of security (1). Balija et al. [10] introduce a peer-to-peer (P2P) federated LLM, namely PageRank, which works with a blockchain. This system operates in a fully decentralized capacity. Demonstrably, the blockchain implementation led to more efficient accuracy and latency results. With that being stated, Balija et al. [10] provide a developing direction in the field of BC4LLM to enhance system security. Below, we address several current vulnerabilities in LLMs and analyze them individually. In order to better understand these security problems, we categorize these vulnerabilities to their respective LLM training stages, highlight blockchain for AI works, and provide well-researched blockchain applications as a solution.

Vulnerabilities are present at each step in the process of developing a LLM. In early methods of model training, we encounter adversarial attacks such as data poisoning and backdoor attacks within the corpus [99; 127]. Progressing into model fine-tuning and general use, the LLM can fall victim to prompt-based attacks [140; 2; 127; 68], inference attacks [55; 44], and RAG-related attacks [126; 22; 5; 26; 149; 132]. These attacks are common vulnerabilities in both LLMs and AI since LLMs and AI are closely related as seen in Figure 2. Some of the threats against LLMs can be addressed by implementations from blockchain for AI (BC4AI) research, as elaborated below in Section *4.1.3*. Considering the volume of potential attacks against LLMs, we make a further distinction of solutions that are specifically related to BC4LLM research and other blockchain-based solutions from BC4AI research. With this, we are able to highlight shared vulnerabilities for LLMs and AI. We provide an analysis of how blockchain can help defend against and mitigate these vulnerabilities, starting with threats during each phase of LLM training and utilization, continuing onto different blockchain solutions for AI inherent threats, and lastly noteworthy technology inherent attacks such as denial of service (DDoS) attacks and issues with supply chain logistics.

4.1.1 Blockchain for Threats in LLMs' Training

To mitigate the threats in LLMs' training, data selection stands as a crucial aspect of model development, with particular emphasis on ensuring that training data is authentic, safe, and resistant to data poisoning attacks. One potential approach could be enabling the LLM to "unlearn" poisoned data or data deemed unsafe based on our definition of safety 2. Zuo et al. [150] establish federated TrustChain to enhance LLMs' training and unlearning through a blockchain-based federated learning framework. Through integration with Hyperledger, the framework can efficiently perform unlearning, reducing the accuracy to 0.70% after unlearning given that the initial accuracy is 99.15% [150]. This demonstrates the potential of applying blockchain techniques to improve the security and privacy of LLMs, where LLMs can selectively forget specified data points while simultaneously preserving the performance via Low-Rank Adaptation (LoRA) and tuning hyper-parameters. This method of a blockchain-enabled federated unlearning process is further detailed as a future research possibility that has been thoroughly explored by few, as emphasized later in Section 7.1.

Another significant issue is data poisoning. To address this issue, Gong et al. [36] propose a possible blockchain solution, introducing dynamic large language models (DLLM) on blockchains. Instead of using the traditional centralized datasets that LLMs are provided with, developing LLMs on blockchains enables the creation of decentralized datasets. These datasets are less likely to be tampered with and can be easily audited for accuracy. Gong et al. [36] present the DLLM to evolve after the training process. This was implemented by adjusting neural network parameters, enabling the LLM to continue learning during its use.

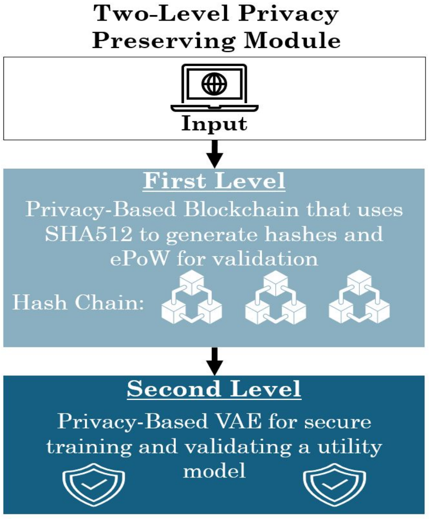

Two-Level Privacy Preserving Module
Input
First Level
Privacy-Based Blockchain that uses SHA512 to generate hashes and ePoW for validation
Hash Chain:
Second Level
Privacy-Based VAE for secure training and validating a utility model

Figure 4: An overview of a two-level framework, consisting of a privacy-based blockchain that uses the secure hash algorithm 512 (SHA512) to generate hashes for data integrity. Then, the enhanced proof of work (ePoW) is used to authenticate data records and prevent data poisoning attacks from altering original data. These hashes of data blocks are linked to each other, called a Hash Chain. Then, for the second level, a privacy-based variational autoencoder (VAE) for secure data transformation ensures robust protection against inference attacks while maintaining the utility model for anomaly detection.

Additionally, blockchain-based systems can help assess where data poisoning may occur, and as shown in [55], blockchain can protect datasets and detect potential inference attacks through a two-level privacy preserving module. This research proposes a framework based on blockchain and deep learning, including two levels of privacy mechanisms as shown in Figure 4. For the first level, Keshk et al. [55] use SHA512 to generate secure hashes and then implement an enhanced-proof-of-work (ePoW) technique for authenticating and preventing data poisoning attacks. The second level consisted of a VAE model for converting original data into an encoded format for mitigating inference attacks that could be learned from system-based machine learning. In their testing, these mechanisms were effective in preventing data poisoning and inference attacks from manipulated smart power network datasets. BC4LLMs could benefit from this similar type of implementation, working with secure methods of hashing and blockchain-based deep learning privacy preservation techniques. By integrating a two-level privacy-preserving module, BC4LLMs can ensure data integrity and confidentiality while effectively detecting and mitigating both data poisoning and inference attacks.

Poisoned data has been an interest with RAG in particular. For example, Xue et al. [126] develop a way to identify security vulnerabilities from a poisoned corpus, but they do not use blockchain as a solution. We address the absence of research on blockchain and RAG, especially when using blockchain to help prevent RAG security issues. We discuss this as a possible future research direction as there remains a current gap in research of blockchain-based RAG systems and elaborate on this topic in Section 7.2. Poisoned data overall is a major concern within LLMs and we offer blockchain as a potential source of ground truth to aid in mitigating this threat during the pre-training stages of LLMs and potentially mitigate RAG security concerns. In addition to data poisoning, LLMs are susceptible to backdoor attacks hidden in the training data during the LLM pre-training phase. Zhao et al. [140] introduce ProAttack which improves the stealth of backdoor attacks by accurately labeling poisoned data samples. As these attacks improve and become more sophisticated, it is crucial to explore robust defense mechanisms for LLMs. Few defense mechanisms using blockchain techniques have been studied, while Li et al. [57] propose a blockchain-based federated-learning framework (DBFL) that withstands backdoor attacks in a blockchain environment by incorporating an RLR aggregation strategy into the aggregation algorithm of a user and the addition of gradient noise to limit the effectiveness of backdoor attacks. The robustness of FL against backdoor attacks is enhanced by using various blockchain functions, including digital signature verification and simulation of chain resynchronization [57].

4.1.2 Blockchain for Threats in LLMs' Prompting and Utilization

LLMs are often further trained through techniques such as instruction tuning, alignment tuning, and fine-tuning, each of which may introduce specific vulnerabilities, such as prompt injection [77; 101] and backdoor attacks [57]. These adversarial attacks are included under the term active privacy 3 where a malicious user attempts to gain unauthorized access. Blockchain technology, with its inherent transparency and immutability, holds the potential to mitigate and defend against these vulnerabilities. For instance, blockchain technology can be used to defend against prompt injections by ensuring data integrity and traceability. Mbula et al. [77] provide an overview of LLMs for blockchain, highlighting how blockchain's transparency and immutability enable a reliable audit trail for tracking and investigating suspicious activities. While not specifically focused on prompt injection, this approach demonstrates how blockchain can enhance security by providing a transparent and immutable record of interactions. Applying this to BC4LLMs can help prevent suspicious users from continuously interacting with an LLM, allowing for traceability to stop the user from entering malicious prompts. We recognize prompt injection is a critical vulnerability in LLMs and AI-related systems, yet as noted in [101], few blockchain defenses for prompt injection are present in the current field of research.

Inference attacks are also a critical concern for LLMs and active privacy, as malicious users may attempt to extract sensitive data from the model, hence why the Taxonomy 3 has inference attacks standalone. As discussed previously in Section *4.1.1*, Keshk et al. [55] apply Blockchain and DL techniques to preserve privacy and prevent inference attacks through a framework as depicted in Figure 4. For more inference attack applicable work, a survey [44] thoroughly discusses membership inference attacks on ML and provides a group of defenses including differential privacy, regularization, confidence masking, and knowledge distillation. In other related works, there are instances of blockchain-based differential privacy methods [143; 37; 87], but current research that uses blockchain-based differential privacy frameworks to prevent inference attacks is limited; It is worth noting that the theoretical foundation and potential synergies of this combination are promising. Another area with limited research is blockchain as a defense for jailbreaking attacks, which exploit the inherent capabilities of LLMs to bypass restrictions. There are multiple articles defending LLMs from jailbreaking attacks, yet little to none fully include blockchain to prevent jailbreaking. Hu et al. [45] explores a blockchain defense mechanism for malware checking on operating systems, indicating a possible direction for future research in integrating blockchain to defend against jailbreaking in LLMs. As previously explained in Section 3.1, backdoor attacks after the model has been trained are based on modifications to the trained model. The key blockchain-based federated-learning framework from Li et al. [57] discussed in detail in Section *4.1.1* used a combination of a blockchain environment and an RLR aggregation strategy to defend against backdoor attacks. This framework effectively coordinated FL processes and maintained learning security and user privacy. When testing backdoor attacks caused by malicious participants, the accuracy of the model increased when using the RLR aggregation strategy [57]. Given these findings, the possibility of leveraging blockchain transparency and immutability presents a robust mechanism for improving LLM security against active privacy threats. However, comprehensive integration and empirical validation of blockchain-based defenses in LLMs remain imperative to advance the field of BC4LLMs.

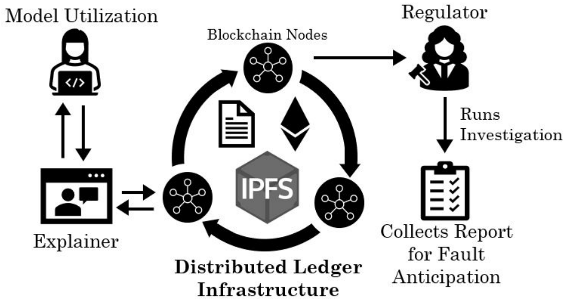


Figure 5: Within BXAI, this diagram illustrates the framework that leverages a distributed ledger infrastructure, using the Ethereum Blockchain and IPFS for storage and secure, traceable transactions. Depicted is the interaction between model utilization: an explainer generating local post-hoc explanations, the storage of these explanations in IPFS, and their linkage to the Blockchain. The use of smart contracts helps secure and encrypt the data, then relay it to a regulator who investigates current explanations, ensuring accountability and fault anticipation. Blockchain nodes are used to facilitate the secure and transparent broadcast of events within the Ethereum network.

4.1.3 AI-intrinsic Threats and Defenses

AI intrinsic threats apply to LLMs due to the proximity of LLMs and AI, as shown in Figure 2. Blockchain for AI (BC4AI) is an emerging technology, with blockchain-based solutions already being researched as a secure way to establish trust in the Internet of Things (IoT) [103; 24]. Before BC4AI, some previous works refer to the integration as "On-chain AI" [28; 23]. Research of BC4AI encapsulates other machine learning techniques, such as blockchain-based federated learning and blockchain for deep learning. Federated Learning is an addition to machine learning, as noted in Figure 2, where federated learning uses a privacy-preserving and decentralized approach to centralized systems.

A substantial literature on BC4AI has emerged from 2018 to 2024 [28; 116; 70; 124; 30; 94; 109; 74; 18; 11; 13]. Among the most notable works, Salah et al. [94] state the integration benefits of BC4AI. For example, there are five main benefits, such as enhanced data security, improved trust in robotic decisions, collective decision-making, decentralized intelligence, and high efficiency [94]. For enhanced data security, information stored within a blockchain is considered highly secure. By storing sensitive and personal data in a distributed, disk-less environment, blockchain can work alongside AI algorithms to strengthen data protection and promote more trusted and credible decision outcomes. The other benefits of improving trust within AI decision-making involve using the blockchain as a record of the decision-making process, allowing better AI traceability to analyze the quality of responses. Secondly, Dinh and Thai [28] summarize the integration of blockchain and AI to where blockchain can assist AI in multiple aspects, as follows. AI can benefit in secure data sharing from blockchain, allowing transparency and accountability regarding which user's data is accessed, when, and by whom, letting users maintain control of their personal data. Among other data concerns, with the integration of blockchain and AI, blockchain technologies allow users to trade data via smart contracts, enabling the possibility of data marketplaces without a centralized middleman, making the transactions private and secure between users. Besides, Malhotra et al. [74] propose a blockchain-based proof-of-authenticity framework for explainable AI (XAI) utilizing a public Ethereum Blockchain, smart contracts, and IPFS (Interplanetary File System) to ensure secure, traceable, auditable transactions within the Ethereum network. This framework highlights three major components, smart contracts, an Ethereum and IPFS interconnected network, and a regulator, as depicted in Figure 5. Using smart contracts can enable continuous monitoring and tracing by all peers, in the case of any rule violations there are prompt rebound transactions to restore the system to an optimal state. To address the size limitation of storage on the blockchain, as further discussed in Section 6.1, Malhotra et al. [74] apply unique IPFS hashes stored on the Ethereum Blockchain to access larger-sized explanations that are stored off-chain in IPFS. These hashes are encrypted with the SHA256 algorithm to maintain data security. Thus, only entities with the corresponding hash can access and retrieve the IPFS hash and the associated explanation, ensuring controlled access even in a distributed network. Lastly, the regulator's role is responsible for auditing and has access to the explanations to predict the user at fault using audit trails if system failure were to occur.

4.1.4 Non-AI Threats and Defenses

Referring to Figure 3, we specifically focus on DDoS attacks and supply chain issues. Even though these attacks are common problems, we consider threats relevant to BC4LLMs. Ibrahim et al. [48] suggest using a public blockchain to prevent DDoS attacks on IoT devices. Blockchain provides a tamper-proof platform as well as demonstrates how IoT devices working with blockchain can verify and authenticate using a trusted white-list which is implemented in the smart contract. Following this smart contract usage, if LLMs were to use a trusted white list for users then we can try to prevent these malicious users from trying to access the LLM in certain circumstances that are mutually agreed upon. Additionally proven by Shah et al. [97], blockchain-based solutions play a vital role in mitigating DDoS attacks.

A point of consideration is how DDoS attacks that target the blockchain to make the blockchain unavailable would require sufficient computer resources. The fully decentralized architecture of the blockchain and the consensus protocol for new blocks ensure that the blockchain can still operate meanwhile several blockchain nodes could be offline [135]. Incorporating this architecture into LLMs would help prevent DDoS attackers, as the larger the blockchain network is, then the harder it would be for a DDoS attack to be successful. Moreover, blockchain is known as a distributed, immutable, and verifiable ledger technology that ensures transparency and traceability [93]. By utilizing blockchain for LLMs, we can help mitigate these supply chain vulnerabilities. The decentralization of the network can maintain the integrity of the system at all points, aiding in mitigating the risk of a single point of failure, a common problem with centralized systems [93]. Blockchain is offered as a solution if the LLM were to accidentally crash, or was purposefully attacked by an attempt at overwhelming the system, then the LLM would still be intact since it is blockchain-based, removing the single point of failure entirely. However, it is important to note that blockchain solutions for LLMs depend on the availability of the underlying LLM infrastructure. If the LLM server is malfunctioning or shuts down, then these blockchain mechanisms may not be applicable, highlighting the need for a robust and resilient supply chain. To solidify the supply chain, blockchain offers secure transactional data in sectors including supply chain management, healthcare, and federated learning [150]. For better supply chain management and data traceability, Kayikci and Khosgoftaar [54] address the potential intersection of blockchain and ML. ML can aid in analyzing data from multiple sources and identify potential supply chain issues such as delays or quality issues before they occur. By using blockchain to create a transparent record of all supply chain transactions there are improvements in security, openness, traceability, and productivity [54]. While blockchain presents a promising solution for enhancing security and defending against adversarial threats to LLMs, ongoing research and development are necessary to address the evolving landscape of threats and vulnerabilities.

4.2 Blockchain for LLMs' Safety

The growing dominance of LLMs as search engines [90], code writers [145], and in many other roles has introduced unique challenges related to their safety. For instance, LLMs who advise users to engage in dangerous activities such as eating glass [39] or which easily reveal personally identifying information [56] may be unsafe for users to interact with even in the absence of external threats. In this section, we rely on our proposed definition of safety (Definition 2) to explore relevant literature that incorporates blockchain technology into the various solutions surrounding LLM safety.

4.2.1 Blockchain for Passive Privacy in LLMs

Despite its novelty, the concept of passive privacy is crucial for ensuring the safety of LLMs. Some models risk leaking sensitive information, potentially exposing private data like government-issued ID numbers and patient records [86]. The severity of these leaks underscores the need for effective solutions to advance LLMs responsibly. In this regard, blockchain's guarantees of data sovereignty, obfuscation, and traceability offer practical passive privacy benefits that align well with the requirements of LLMs. In particular, we observe blockchain-based privacy preservation techniques which originate in varying proximity to LLMs as seen in BC4LLMs itself [113; 118; 102], blockchain-enabled deep learning [147; 55; 121; 120; 96], blockchain-enabled machine learning [7], and blockchain-enabled federated learning [8; 81; 142; 88; 31; 89; 73].

Within our focus on BC4LLMs, we have observed distinct trends in the application of blockchain to LLMs in their capacity to bolster passive privacy guarantees. Most notably, the development of zero-knowledge LLMs, a.k.a. ZK-LLMs, as described in [118] and [102], has the potential to drastically reduce privacy leakage risks when interacting with LLMs. Considering the problem of data leakage approached from the lens of access, this application is natural. A user querying for their own personally identifiable information should, ideally, be able to access it whereas an unauthorized user should not. Obfuscating portions or the entirety of a corpus using zero-knowledge proofs [133] allows for untrusted training nodes, or the model itself, to act on sensitive data without the ability to regurgitate it to a potentially malicious party. This same mechanism has broad applications that have been explored in other recent works as well, with special focuses on ZKPs for data curation and pre-processing [113], which consequently enhance passive privacy within LLMs.

Additionally, besides material on BC4LLMs, it is necessary to discuss passive privacy contributions made within LLM-related areas, as described in our classification of LLMs in the context of AI, ML, and DL (Figure 2). Especially important in its immediate applications to LLMs, blockchain-enabled deep learning (BC-DL) is a growing field with potentially large impacts on LLM's passive privacy. Specifically, certain BC-DL technologies propose learning mechanisms distinct from traditional federated learning models [55; 121; 96]. The concerted research effort to develop efficient distributed learning models that deviate from the typical model of federated learning is clearly well underway. This field has broad implications for blockchain; through the utilization of various blockchain properties, we see the development of privacy guarantees which undoubtedly strengthens the BC4LLMs area.

A noteworthy contribution in the field of blockchain and deep learning is the influential DeepChain [121], which introduces a novel privacy-preserving training framework based on blockchain technology. This system employs a consensus protocol alongside an incentive mechanism, enabling the use of private training gradients and ensuring the auditability of training data. This dual approach, incorporating zero-knowledge proofs in various aspects of the protocol, represents a promising new direction for blockchain-enabled passive privacy, building upon the well-established domain of federated learning. To underscore the novelty of this approach, Figure 6 illustrates how the system operates primarily through consensus and incentivization mechanisms.

Additionally, despite this potential research direction, discussion on the wide body of research that does exist concern-

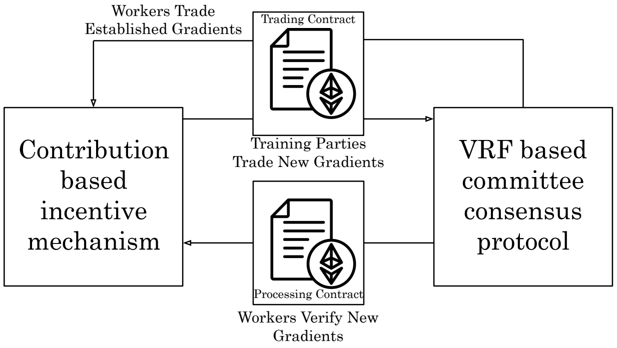


Figure 6: DeepChain deviates from conventional federated learning models by incorporating a synchronous requirement for consensus on data incorporated into the model during the final training round. The contribution-based incentive mechanism rewards participants for verifying new gradients and activates a trading contract to facilitate the sharing of updated gradient information. Besides, a Verifiable Random Function (VRF) ensures fairness in the committee-based consensus process, addressing concerns related to finality.

ing blockchain-enabled federated learning (BC-FL) is essential when describing blockchain as a vehicle for improving the safety and security of LLMs. Federated learning, introduced by McMahan et al. [78], has since been the principal building block of decentralized learning approaches in machine learning systems. While too broad to be considered for BC4LLMs, notable to this paper are the contributions of authors approaching BC-FL in its capacity as a powerful privacy preserving mechanism [81; 8; 142; 31; 32; 73].

4.2.2 Blockchain for Misinformation in LLMs

The issue of LLMs fabricating information, commonly known as hallucinations [75; 3] is well understood. As a result, detecting and defending against hallucinations is a widely explored area [95], with more research still yet to be conducted [131; 136]. Along with this pressing general body of research, efforts have been made to leverage blockchain technology to reduce hallucinations by consensus-oriented [138] and oracle-based [15] approaches. Within consensus, Zhang et al. [138] proposed a system for efficient large language model inference quality assessment. That is, the veracity of a given model's responses was able to be assessed by using a 'Proof of Quality' consensus mechanism with low latency between the user and language model. This stands in contrast to other approaches, such as Bouchiha et al.'s [15] reputation-based system LLMChain, which relies on a decentralized oracle to cross reference request/response pairs originating from differing models and speak to the quality of inferences based on those comparisons. It is worth noting that despite these fundamental differences, they are both consensus-based approaches. This serves as an excellent example of how BC4LLM technology can take many different forms towards the same goal.

In addition to advancing consensus-driven governance models for mitigating misinformation, several approaches have been explored to improve the accuracy of LLMs' responses. These efforts include contributions from the zero-knowledge domain, as demonstrated by Chen et al. [21], more optimistic privacy assurances found in works like [23], and straightforward applications of verifiable ledgers, as proposed by Yazdinejad et al. [130]. Chen et al. [21] propose zkML, a compiler that enables TensorFlow models to be translated readily into zk-SNARK halo2 circuits via either KZG or IPA commitments. This conversion allows for any portion, or the entirety of, an LLM to gain the properties of zero-knowledge, knowledge soundness, and completeness. Through this, and with potential connections to verifiable databases, zkML gains the powerful ability to audit inferences and ensure their accuracy. This research avenue is particularly promising due to both the efficient and potentially on-chain verification of zk-SNARKs as well as the extensibility of zkML to virtually any ML model.

Distinct from the zkML approach is opML [23], which opts for an optimistic approach reliant on a fraud-proof rather than a ZK proof to catch erroneous outputs within a certain challenge period. Clearly, there exist trade-offs in this implementation when compared to the zkML approach. Optimistic rollups are desirable in the sense that they are performant, but if implemented in a RAG environment, or similarly situated between the user and a model, latency issues can quickly become dominant. Apart from proof-oriented mechanisms and worth noting is the work proposed by Yazdinejad et al. [130], which focuses on detecting deepfakes using blockchain's verifiable ledger. While not directly applicable to the realm of LLMs due to the non-atomic nature of data within a language model, important insights can be drawn from the paper. Namely, BC4LLMs could benefit greatly from a proposed hashing method applied to particularly sensitive data areas such as names, addresses, or even health-care-related parts of corpora. This hash could be used as a guarantee of data veracity and could potentially prevent unsafe behaviors such as sycophancy, deception, or unfairness. Indeed, this hashing mechanism has the potential to be used as a final check for the LLM to verify that it is submitting information to the user that is consistent with standards agreed upon when information was originally committed to the ledger. Many similar vehicles for the maintenance of data integrity exist, albeit currently limited by scaling issues on-chain [27].

5. DATASETS RELEVANT TO BC4LLM

Developing synergistic techniques that integrate blockchain with LLMs is essential for ensuring the safety and trustworthiness of future LLMs. In this context, it is critical to access to relevant datasets for experimentation. Blockchain-enabled systems often require unconventional training sets and edge cases to capture the dynamism and robustness of these implementations. Accordingly, we have compiled and summarized the relevance of specific datasets in Table 4. While we include standard datasets such as MNIST, CIFAR-10, SQuAD, and MS-MARCO, we also highlight lesser-known datasets that may prove valuable in particular research contexts.

6. CHALLENGES IN BC4LLM

Despite the promise of the emerging BC4LLMs field, there are several innate challenges that delay progress and inhibit potential research directions. Typically, these are derived

Table 4: **Datasets Relevant to BC4LLMs**

Dataset	**Use Case**	**Description**	**Papers**
MNIST[1]	Pattern Recognition	Images of handwritten digits for pattern recognition applications or vulnerability analysis.	[121; 119; 78; 31; 100; 58]
CIFAR-10[2]		Labeled images used in capacities from improving pattern recognition to zk-SNARK benchmarks.	[119; 21; 78]
MS MARCO[3]		Collection of human answered questions, used in training corpora as well as simulating RAG attacks.	[126; 149; 22]
MedMINST[4]		Collection of medical images from case studies.	[126; 149; 22]
Natural Questions[5]	Poisoned RAG	Open domain question answering dataset, incorporating questions from users and rigorous answers.	[126; 149; 22]
HotpotQA[6]		Question answering dataset with multi-hop questions and supervised, regulated, answers.	[149; 22]
MT BENCH[7]	LLM Evaluation	Ranked pairwise expert human preferences for various model responses.	[144; 15]
SQuAD[8]		Reading comprehension dataset comprised of questions posed on Wikipedia article with answers as sections of those corresponding articles.	[150; 46]
IMDB Dataset[9]	Sentiment Analysis	Movie reviews	[150; 46]
SafetyBench[10]	Safety Evaluation	Large number of multiple choice questions focused on evaluating the safety of LLMs.	[137]
Tweets2011[11]	Sensitive Information Handling	List of scraped tweet identifiers and corresponding tweets from early 2011.	[150]
MTSamples Scrape[12]		Sample transcription medical reports from various disciplines and areas.	[150; 46]
DRC Diplomas[13]		Highschool diplomas from the Democratic Republic of the Congo.	[10]
HealthCareMagic[14]		Real patient-doctor conversations found through the HealthCareMagic website, capturing the nature of patient vocabulary.	[5]
Enron Emails[15]		Large set of emails generated by employees of the Enron Corporation.	[150; 46]
LLMGooAQ[16]		Comprehensive database capturing question and answers from a wide variety of domains.	[15]
GooAQ[17]		Large scale question answering dataset aimed at developing a vast selection of question types.	[15]
The Pile[18]		Massive and open source data set consisting of a combination of roughly 20 other datasets.	[140]

[1] https://yann.lecun.com/exdb/mnist/ [2] https://www.cs.toronto.edu/ kriz/cifar.html [3] https://microsoft.github.io/msmarco/
[4] https://medmnist.com/ [5] https://ai.google.com/research/NaturalQuestions [6] https://hotpotqa.github.io/
[7] https://paperswithcode.com/dataset/mt-bench [8] https://rajpurkar.github.io/SQuAD-explorer/
[9] https://developer.imdb.com/non-commercial-datasets/ [10] https://github.com/thu-coai/SafetyBench
[11] https://trec.nist.gov/data/tweets/ [12] https://mtsamples.com/ [13] https://minepst.gouv.cd/palmares-exetat/
[14] https://huggingface.co/datasets/RafaelMPereira/HealthCareMagic-100k-Chat-Format-en
[15] https://huggingface.co/datasets/preference-agents/enron-cleaned [16] https://github.com/mohaminemed/LLMGooAQ/
[17] https://huggingface.co/datasets/allenai/gooaq [18] https://pile.eleuther.ai/

from certain limitations in blockchain technology, LLMs, or deficits in the way that blockchain can serve LLMs.

6.1 Corpus on Blockchain

LLMs' training heavily relies on substantial data, with modern corpora typically exceeding dozens of terabytes in volume [141]. This characteristic is inherently misaligned with the constraints that blockchain systems are generally designed to address. Reconciling blockchain's limitations in throughput and data-handling capacity with the extensive data requirements of LLMs represents one of the most pressing challenges in BC4LLMs. Several approaches have explored the use of zero-knowledge proofs (ZKPs) to enhance scalability [118; 102]. However, relying on zero-knowledge technology solely for scalability, and not privacy, poses challenges due to the computational cost of generating ZKPs, even with minimal circuits. A significant factor here is the ongoing issue of the Multi-Scalar Multiplication (MSM) in ZKP generation [125]. Furthermore, current WebGPU and WASM implementations likely fall short of the throughput required by client-based LLMs. For these reasons, it is improbable that zero knowledge could serve as a definitive solution to scalability in BC4LLMs without significant advancements in zk-SNARK generation research. Addressing the discrepancy between the growing size of LLMs and blockchain's limited capacity for on-chain data storage remains a substantial research challenge.

6.2 Reliance on Oracles

The security guarantees of blockchain technology, while robust, often leave little room for interoperability with external systems [27]. That is, the blockchain can most easily interact with information on the chain, leaving little room to consider issues such as fact-checking or moral alignment. Oftentimes, to develop mechanisms that seek to assist with LLM toxicity or factuality, oracles are used to bridge this gap [33; 30]. Serving as mediators between chains and online sources, oracles are trusted parties that deliver information through a variety of protocols and frameworks. However, introducing a trusted party into an otherwise trustless system has been a long-standing weak point in this solution [77]. Exploring non-oracle-based options for ground truth solutions, or toxicity checks, would greatly enhance the security guarantees of blockchain within LLMs.

6.3 Energy Consumption

A significant portion of the challenges associated with LLMs arises from their need to consume and process vast amounts of data [141]. This requirement, in turn, necessitates extensive energy consumption during both training and runtime [72]. On the other hand, blockchain systems face their own energy challenges, as consensus mechanisms and transaction validation processes often incur substantial computational costs [77]. The high computational demands of both LLMs and blockchain systems highlight a misalignment with the scalability of BC4LLM implementations without substantial efforts to reduce energy costs. This might require moving away from transformer-based architectures and energy-intensive consensus mechanisms, such as proof of work, toward more sustainable alternatives [84].

7. FUTURE RESEARCH DIRECTIONS

There exist several critically overlooked areas within LLMs that may benefit greatly from the introduction of blockchain technologies. The most prominent of these areas include blockchain federated unlearning, RAG, differential privacy, data provenance, and toxicity mitigation.

7.1 Blockchain Federated Unlearning

Privacy regulations are paramount in the online realm, especially concerning the "right to be forgotten" and user data privacy which are critical considerations when working with LLMs and blockchain. Federated blockchain unlearning offers LLMs the ability to erase learned data. Within our research, we identified four recent papers that have implemented blockchain-based federated unlearning frameworks. As noted previously in Section *4.1.1*, Zuo et al. [150] develop a federated TrustChain framework for blockchain-enhanced LLM training and unlearning, focusing on the impact of Low-Rank Adaptation (LoRA) hyperparameters on unlearning performances and integrating Hyperledger Fabric to ensure the security, transparency, and verifiability of the unlearning process. In another study, Zuo et al. [151] presented a trustworthy approach towards federated learning with blockchain-enhanced machine unlearning. This implementation differs from Trustchain, where Zuo et al. [151] used a machine unlearning mechanism that utilized two types of clients for training and unlearning, smart contracts for process automation, and a blockchain network for secure, immutable record-keeping. Beyond the above works [151], Liu et al. [66] introduced Blockchain Federated Unlearning (BlockFUL) as a versatile framework that redesigns the blockchain structure using a Chameleon Hash (CH) technology to simplify model updates and reduce the computational and consensus costs associated with unlearning tasks. Additionally, BlockFUL ensures the integrity and traceability of model updates, including privacy-preserving results from these blockchain-based unlearning operations [66]. Lin et al. [63] propose a framework with a proof of federated unlearning protocol that also utilizes the Chameleon hash function to verify data removal and eliminate the data contributions stored in other clients' models. Both use CH functions in their blockchain-enabled federated unlearning processes. The applications of key blockchain components, such as on-chain smart contracts and hash mappings for verifying data removal, may enable LLMs to forget personal data effectively. Blockchain for unlearning is an emerging area of research with significant potential for further innovation.

7.2 Blockchain-enhanced RAG

Considering the popularity of Retrieval Augmented Generation (RAG), extensive studies have emerged to focus on potential vulnerabilities within RAG that could compromise the integrity of LLMs [126; 5; 26; 149; 46; 132; 22]. However, a significant gap remains in addressing strategies to mitigate these attacks, particularly where blockchain technology could offer defensive benefits. Recent studies have called for exploring blockchain's role in RAG deployment [9], and preliminary investigations have assessed blockchain's potential to enhance user experience [134] and performance evaluation [87]. Nonetheless, dedicated efforts to strengthen security and safety within RAG systems are largely absent in the current literature. Advancing BC4LLMs specifically in the context of RAG security could yield considerable mutual benefits for both blockchain and LLM technologies.

7.3 Blockchain for Privacy Guarantees in LLMs

The clear connection between federated learning, blockchain, and LLMs allows for the field of differential privacy to enter BC4LLMs' sphere of relevance. Major contributions concerning the impact of differential privacy on related areas such as deep learning have already been made [1], but issues such as privacy budget exhaustion still loom large in the space [14]. Moreover, despite conclusions that blockchain can help with privacy budget exhaustion [37; 143], few efforts have been conducted in exploring these solutions. Indeed, there is a need for more relevant research in order to realize the full measure of blockchain's impact on this area.

7.4 Blockchain for Data Provenance and Transparency in LLMs

Several recent papers have urged for increased data accountability measures to be placed on organizations developing LLMs, especially where it concerns issues of data acquisition [12; 40]. Additionally, worth noting are direct calls for the introduction of blockchain technology to help solve the issue of data provenance [122] in LLMs. Largely, while this has been answered with responses in the realms of auditability [57], straightforward data tracking solutions have remained absent from the literature, despite relatively simple conceptual formulations [28]. Towards this goal of achieving improved data provenance within LLMs corpus', RAG databases, and even in-context learning repositories, there is a need for more explorations into this natural application of distributed ledger technology to problems of explainable AI concerning LLMs.

7.5 Blockchain for Non-toxic LLMs

Encompassing vital attributes such as ethics, legality, and non-violence, developing non-toxic LLMs has been and will continue to be a major focus of the field for the foreseeable future [101; 69; 38]. There is no doubt that automated filtering of generated toxic content is one of the most pressing challenges concerning the safety of LLMs [35; 6]. This is because in essence, filtering inferences negatively impacts the quality of LLM responses, whereas manual human annotation is a costly and complex process [6]. Therefore, the applications of blockchain technology in this regard, while currently limited, are compelling. Considering one of the most groundbreaking achievements in ML within the past several years, federated learning has allowed for massive strides to be made within the spaces of securing training sets, user privacy, and even misinformation defense. A similar approach, aimed at the problem of toxicity, could be a hugely beneficial endeavor to the field. Moreover, imagining such a model is not difficult. Developing consensus around what is considered correct in a model and using that to propagate gradients and parameters is not dissimilar to the decisions that must be made about what is or is not toxic given the state of certain corpora. Given a concentrated research program, automated non-toxicity could very well have excellent solutions found in the blockchain space.

8. CONCLUSION

In this survey, we first highlight significant systemic vulnerabilities in large language models (LLMs), including data poisoning, hallucinations, jailbreaking, and privacy attacks. Although these issues have been extensively studied in conventional machine learning models, with approaches like differential privacy and federated learning, comprehensive protection for LLMs remains an area for improvement. In contrast, blockchain technology offers a promising solution to enhance the security and safety of LLMs. Blockchain systems provide powerful mechanisms to ensure data integrity, provenance, and encrypted frameworks, which can be leveraged to strengthen LLM defenses. By integrating blockchain-based defenses, it is possible to achieve stronger privacy protection, reliable data, and improved resilience of LLMs against adversarial threats.

Besides, it is critical to establish clear definitions of security and safety in the context of LLMs. We conclude that security for LLMs pertains to the ability to tolerate applicable adversarial attacks while maintaining system integrity to provide consistent and accurate responses, whereas safety for LLMs is the model's capacity to interact with users in a trustworthy manner, contingent upon adhering to ethical concerns, law-abiding, non-violent, fair, passively privacy-preserving, and informing. Additionally, differentiating between active and passive privacy measures will aid in developing more targeted and effective privacy-preserving strategies. These distinctions and definitions provide a foundational framework for future research in BC4LLM. From analyzing the integration of blockchain and LLMs, we propose a new taxonomy in Figure 3, where previous research done in the field of BC4LLMs can apply to security and safety problems that LLMs face. We recognize various gaps in BC4LLMs that need to be looked into for further consideration. In refining our understanding of relevant concepts, we see that the intersection of blockchain and LLMs holds significant potential for addressing the current shortcomings in LLM security and safety. Through our review, we aim to guide new researchers in understanding how blockchain technology can be utilized to enhance the security, reliability, and safety of LLMs.

9. REFERENCES

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, page 308–318, 2016.

[2] S. Abdali, R. Anarfi, C. Barberan, and J. He. Securing large language models: Threats, vulnerabilities and responsible practices. *arXiv preprint arXiv:2403.12503*, 2024.

[3] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024.

[4] O. Ali, A. Jaradat, A. Kulakli, and A. Abuhalimeh. A comparative study: Blockchain technology utilization benefits, challenges and functionalities. *IEEE Access*, 9:12730–12749, 2021.

[5] M. Anderson, G. Amit, and A. Goldsteen. Is my data in your retrieval database? membership inference attacks against retrieval augmented generation. *arXiv preprint arXiv:2405.20446*, 2024.

[6] U. Anwar, A. Saparov, J. Rando, D. Paleka, M. Turpin, P. Hase, E. S. Lubana, E. Jenner, S. Casper, O. Sourbut, et al. Foundational challenges in assuring alignment and safety of large language models. *arXiv preprint arXiv:2404.09932*, 2024.

[7] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman. A trustworthy privacy preserving framework for machine learning in industrial iot systems. *IEEE Transactions on Industrial Informatics*, 16(9):6092–6102, 2020.

[8] S. Awan, F. Li, B. Luo, and M. Liu. Poster: A reliable and accountable privacy-preserving federated learning framework using the blockchain. pages 2561–2563, 2019.

[9] A. Balakrishnan. Enhancing data engineering efficiency with ai: Utilizing retrieval-augmented generation, reinforcement learning from human feedback, and fine-tuning techniques. *International Research Journal of Modernization in Engineering Technology and Science*, 6, 2024.

[10] S. B. Balija, A. Nanda, and D. Sahoo. Building communication efficient asynchronous peer-to-peer federated llms with blockchain. *Proceedings of the AAAI Symposium Series*, 3(1):288–292, 2024.

[11] N. Baranwal Somy, K. Kannan, V. Arya, S. Hans, A. Singh, P. Lohia, and S. Mehta. Ownership preserving ai market places using blockchain. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 156–165, 2019.

[12] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.

[13] D. Bhumichai, C. Smiliotopoulos, R. Benton, G. Kambourakis, and D. Damopoulos. The convergence of artificial intelligence and blockchain: The state of play and the road ahead. *Information*, 15(5), 2024.

[14] A. Bkakria, A. Tasidou, N. Cuppens-Boulahia, F. Cuppens, F. Bouattour, and F. Fredj. *Optimal Distribution of Privacy Budget in Differential Privacy*, pages 222–236. 2019.

[15] M. A. Bouchiha, Q. Telnoff, S. Bakkali, R. Champagnat, M. Rabah, M. Coustaty, and Y. Ghamri-Doudane. Llmchain: Blockchain-based reputation system for sharing and evaluating large language models. *arXiv preprint arXiv:2404.13236*, 2024.

[16] T. B. Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[17] V. Buterin. Ethereum: A next-generation smart contract and decentralized application platform. *white paper*, 3(37), 2014.

[18] D. Calvaresi, Y. Mualla, A. Najjar, S. Galland, and M. Schumacher. Explainable multi-agent systems through blockchain technology. In *Explainable, Transparent Autonomous Agents and Multi-Agent Systems: First International Workshop, EXTRAAMAS 2019, Montreal, QC, Canada, May 13–14, 2019, Revised Selected Papers 1*, pages 41–58. Springer, 2019.

[19] B. Cao, Z. Wang, L. Zhang, D. Feng, M. Peng, L. Zhang, and Z. Han. Blockchain systems, technologies, and applications: A methodology perspective. *IEEE Communications Surveys & Tutorials*, 25(1):353–385, 2023.

[20] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270, 2023.

[21] B.-J. Chen, S. Waiwitlikhit, I. Stoica, and D. Kang. Zkml: An optimizing system for ml inference in zero-knowledge proofs. pages 560–574, 2024.

[22] P. Cheng, Y. Ding, T. Ju, Z. Wu, W. Du, P. Yi, Z. Zhang, and G. Liu. Trojanrag: Retrieval-augmented generation can be backdoor driver in large language models. *arXiv preprint arXiv:2405.13401*, 2024.

[23] K. Conway, C. So, X. Yu, and K. Wong. opml: Optimistic machine learning on blockchain. *arXiv preprint arXiv:2401.17555*, 2024.

[24] J. Cuomo. How blockchain adds trust to ai and iot, 2020.

[25] C. Deng, Y. Duan, X. Jin, H. Chang, Y. Tian, H. Liu, H. P. Zou, Y. Jin, Y. Xiao, Y. Wang, et al. Deconstructing the ethics of large language models from long-standing issues to new-emerging dilemmas. *arXiv preprint arXiv:2406.05392*, 2024.

[26] G. Deng, Y. Liu, K. Wang, Y. Li, T. Zhang, and Y. Liu. Pandora: Jailbreak gpts by retrieval augmented generation poisoning. *arXiv preprint arXiv:2402.08416*, 2024.

[27] A. Deshpande, K. Stewart, L. Lepetit, and S. Gunashekar. Distributed ledger technologies/blockchain: Challenges, opportunities and the prospects for standards. *Overview report The British Standards Institution (BSI)*, 40(40):1–34, 2017.

[28] T. N. Dinh and M. T. Thai. Ai and blockchain: A disruptive integration. *Computer*, 51(9):48–53, 2018.

[29] S. B. ElMamy, H. Mrabet, H. Gharbi, A. Jemai, and D. Trentesaux. A survey on the usage of blockchain technology for cyber-threats in the context of industry 4.0. *Sustainability*, 12(21):9179, 2020.

[30] S. Fan, N. Ilk, A. Kumar, R. Xu, and J. L. Zhao. Blockchain as a trust machine: From disillusionment to enlightenment in the era of generative ai. *Decision Support Systems*, 182, 2024.

[31] A. Ferenczi and C. Bădică. A fully decentralized privacy-enabled federated learning system. In *Computational Collective Intelligence: 15th International Conference, ICCCI Proceedings*, pages 444–456, 2023.

[32] R. Fotohi, F. Shams Aliee, and B. Farahani. Decentralized and robust privacy-preserving model using blockchain-enabled federated deep learning in intelligent enterprises. *Applied Soft Computing*, 161:111764, 2024.

[33] P. Fraga-Lamas and T. M. Fernández-Caramés. Fake News, Disinformation, and Deepfakes: Leveraging Distributed Ledger Technologies and Blockchain to Combat Digital Deception and Counterfeit Reality. *IT Professional*, 2020.

[34] D. Ganguli, L. Lovitt, J. Kernion, A. Askell, Y. Bai, S. Kadavath, B. Mann, E. Perez, N. Schiefer, K. Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

[35] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. pages 3356–3369, 2020.

[36] Y. Gong. Dynamic large language models on blockchains. *arXiv preprint arXiv:2307.10549*, 2023.

[37] L. M. Han, Y. Zhao, and J. Zhao. Blockchain-based differential privacy cost management system. *arXiv preprint arXiv:2006.04693*, 2020.

[38] T. Han, A. Kumar, C. Agarwal, and H. Lakkaraju. Towards safe large language models for medicine. *ICML 2024 Workshop on Models of Human Feedback for AI Alignment*, 2024.

[39] S. Harrer. Attention is not all you need: the complicated case of ethically using large language models in healthcare and medicine. *EBioMedicine*, 90, 2023.

[40] K. He, R. Mao, Q. Lin, Y. Ruan, X. Lan, M. Feng, and E. Cambria. A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics. *arXiv preprint arXiv:2310.05694*, 2024.

[41] Z. He, Z. Li, S. Yang, A. Qiao, X. Zhang, X. Luo, and T. Chen. Large language models for blockchain security: A systematic literature review. *arXiv preprint arXiv:2403.14280*, 2024.

[42] D. Hendrycks, C. Burns, S. Basart, A. Critch, J. Li, D. Song, and J. Steinhardt. Aligning ai with shared human values. 2020.

[43] T. F. Heston. Prespective chapter: Integrating large language models and blockchain in telemedicine. 2024.

[44] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37, 2022.

[45] Q. Hu, M. R. Asghar, and S. Zeadally. Blockchain-based public ecosystem for auditing security of software applications. 103(11):2643–2665, 2021.

[46] Z. Hu, C. Wang, Y. Shu, P. Helen, and L. Zhu. Prompt perturbation in retrieval-augmented generation based large language models. *arXiv preprint arXiv:2402.07179*, 2024.

[47] X. Huang, W. Ruan, W. Huang, G. Jin, Y. Dong, C. Wu, S. Bensalem, R. Mu, Y. Qi, X. Zhao, K. Cai, Y. Zhang, S. Wu, P. Xu, D. Wu, A. Freitas, and M. A. Mustafa. A survey of safety and trustworthiness of large language models through the lens of verification and validation. 57(7), 2024.

[48] R. F. Ibrahim, Q. Abu Al-Haija, and A. Ahmad. Ddos attack prevention for internet of thing devices using ethereum blockchain technology. *Sensors*, 22(18), 2022.

[49] H. Inan, K. Upasani, J. Chi, R. Rungta, K. Iyer, Y. Mao, M. Tontchev, Q. Hu, B. Fuller, D. Testuggine, and M. Khabsa. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

[50] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. Survey of hallucination in natural language generation. 55(12):1–38, 2023.

[51] T. Jiang, Z. Wang, J. Liang, C. Li, Y. Wang, and T. Wang. Robustkv: Defending large language models against jailbreak attacks via kv eviction. *arXiv preprint arXiv:2410.19937*, 2024.

[52] H. Jin, L. Hu, X. Li, P. Zhang, C. Chen, J. Zhuang, and H. Wang. Jailbreakzoo: Survey, landscapes, and horizons in jailbreaking large language and vision-language models. *arXiv preprint arXiv: 2407.01599*, 2024.

[53] E. Kasneci, K. Sessler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, S. Krusche, G. Kutyniok, T. Michaeli, C. Nerdel, J. Pfeffer, O. Poquet, M. Sailer, A. Schmidt, T. Seidel, M. Stadler, J. Weller, J. Kuhn, and G. Kasneci. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103, 2023.

[54] S. Kayikci and T. M. Khoshgoftaar. Blockchain meets machine learning: A survey. *Journal of Big Data*, 11(1), 2024.

[55] M. Keshk, B. Turnbull, N. Moustafa, D. Vatsalan, and K.-K. R. Choo. A privacy-preserving-framework-based blockchain and deep learning for protecting smart power networks. 16(8):5110–5118, 2020.

[56] S. Kim, S. Yun, H. Lee, M. Gubri, S. Yoon, and S. J. Oh. Propile: Probing privacy leakage in large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[57] L. Li, J. Qin, and J. Luo. A blockchain-based federated-learning framework for defense against backdoor attacks. 12(11), 2023.

[58] Z. Li, H. Yu, T. Zhou, L. Luo, M. Fan, Z. Xu, and G. Sun. Byzantine resistant secure blockchained federated learning at the edge. *IEEE Network*, 35(4):295–301, 2021.

[59] J. Liang, S. Li, B. Cao, W. Jiang, and C. He. Omnilytics: A blockchain-based secure data market for decentralized machine learning. *arXiv preprint arXiv:2107.05252*, 2021.

[60] Z. Liang, J. Cheng, R. Yang, H. Ren, Z. Song, D. Wu, X. Qian, T. Li, and Y. Shi. Unleashing the potential of llms for quantum computing: A study in quantum architecture design. *arXiv preprint arXiv:2307.08191*, 2023.

[61] F. Lin, S. Crawford, K. Guillot, Y. Zhang, Y. Chen, X. Yuan, L. Chen, S. Williams, R. Minvielle, X. Xiao, et al. Mmst-vit: Climate change-aware crop yield prediction via multi-modal spatial-temporal vision transformer. In *Proceedings of the IEEE/CVF International*

Conference on Computer Vision, pages 5774–5784, 2023.

[62] F. Lin, X. Yuan, Y. Zhang, P. Sigdel, L. Chen, L. Peng, and N.-F. Tzeng. Comprehensive transformer-based model architecture for real-world storm prediction. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 54–71. Springer, 2023.

[63] Y. Lin, Z. Gao, H. Du, J. Ren, Z. Xie, and D. Niyato. Blockchain-enabled trustworthy federated unlearning. *arXiv preprint arXiv:2401.15917*, 2024.

[64] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin. When machine learning meets privacy: A survey and outlook. *ACM Computing Surveys (CSUR)*, 54(2):1–36, 2021.

[65] M. Liu, S. Ho, M. Wang, L. Gao, Y. Jin, and H. Zhang. Federated learning meets natural language processing: A survey. *arXiv preprint arXiv:2107.12603*, 2021.

[66] X. Liu, M. Li, X. Wang, G. Yu, W. Ni, L. Li, H. Peng, and R. Liu. Decentralized federated unlearning on blockchain. *arXiv preprint arXiv:2402.16294*, 2024.

[67] X. Liu, J. Liang, L. Tang, C. You, M. Ye, and Z. Xi. Buckle up: Robustifying llms at every customization stage via data curation. *arXiv preprint arXiv:2410.02220*, 2024.

[68] Y. Liu, G. Deng, Y. Li, K. Wang, Z. Wang, X. Wang, T. Zhang, Y. Liu, H. Wang, Y. Zheng, and Y. Liu. Prompt injection attack against llm-integrated applications. *arXiv preprint arXiv:2306.05499*, 2024.

[69] Y. Liu, Y. Yao, J.-F. Ton, X. Zhang, R. Guo, H. Cheng, Y. Klochkov, M. F. Taufiq, and H. Li. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*, 2024.

[70] V. Lopes and L. A. Alexandre. An overview of blockchain integration with robotics and artificial intelligence. *arXiv preprint arXiv:1810.00329*, 2018.

[71] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363. IEEE, 2023.

[72] H. Luo, J. Luo, and A. V. Vasilakos. Bc4llm: Trusted artificial intelligence when blockchain meets large language models. *arXiv preprint arXiv:2310.06278*, 2023.

[73] S. K. M., S. Nicolazzo, M. Arazzi, A. Nocera, R. R. K. A., V. P, and M. Conti. Privacy-preserving in blockchain-based federated learning systems. *Computer Communications*, 2024.

[74] D. Malhotra, P. Saini, and A. K. Singh. Blockchain-based proof-of-authenticity frameworks for explainable ai. *Multimedia Tools and Applications*, 83(13):37889–37911, 2024.

[75] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*, 2020.

[76] J. G. M. Mboma, K. Lusala, M. Matalatala, O. T. Tshipata, P. S. Nzakuna, and D. T. Kazumba. Integrating llm with blockchain and ipfs to enhance academic diploma integrity. In *2024 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*, pages 1–6, 2024.

[77] J. G. M. Mboma, O. T. Tshipata, W. V. Kambale, and K. Kyamakya. Assessing how large language models can be integrated with or used for blockchain technology: Overview and illustrative case study. In *2023 27th International Conference on Circuits, Systems, Communications and Computers (CSCC)*, pages 59–70. IEEE, 2023.

[78] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas. Communication-efficient learning of deep networks from decentralized data. *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[79] R. C. Merkle. A digital signature based on a conventional encryption function. pages 369–378. Springer, 1988.

[80] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun. A review on consensus algorithm of blockchain. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2567–2572, 2017.

[81] A. Nagar. Privacy-preserving blockchain based federated learning with differential data sharing. *arXiv preprint arXiv:1912.04859*, 2019.

[82] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[83] S. Neel and P. Chang. Privacy issues in large language models: A survey. *arXiv preprint arXiv:2312.06717*, 2024.

[84] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz. Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals, applications and opportunities. *IEEE Access*, 2019.

[85] A. Pal, L. K. Umapathi, and M. Sankarasubbu. Medhalt: Medical domain hallucination test for large language models. *arXiv preprint arXiv:2307.15343*, 2023.

[86] X. Pan, M. Zhang, S. Ji, and M. Yang. Privacy risks of general-purpose language models. pages 1314–1331, 2020.

[87] Y.-H. Park, Y. Kim, and J. Shim. Blockchain-based privacy-preserving system for genomic data management using local differential privacy. *Electronics*, 10(23):3019, 2021.

[88] A. Qammar, A. Karim, H. Ning, and J. Ding. Securing federated learning with blockchain: a systematic literature review. *Artificial Intelligence Review*, 56(5):3951–3985, 2023.

[89] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood. Blockchain-enabled federated learning: A survey. *ACM Computing Surveys*, 55(4):1–35, 2023.

[90] L. Reid. Generative ai in search: Let google do the searching for you, 2024.

[91] K. Ruan, X. He, J. Wang, X. Zhou, H. Feng, and A. Kebarighotbi. S2e: Towards an end-to-end entity resolution solution from acoustic signal. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10441–10445, 2024.

[92] P. Röttger, F. Pernisi, B. Vidgen, and D. Hovy. Safetyprompts: a systematic review of open datasets for evaluating and improving large language model safety. *arXiv preprint arXiv:2404.05399*, 2024.

[93] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen. Blockchain technology and its relationships to sustainable supply chain management. 57(7):2117–2135, 2019.

[94] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. Al-Fuqaha. Blockchain for ai: Review and open research challenges. *IEEE Access*, 7:10127–10149, 2019.

[95] O. Seneviratne. Blockchain for social good: Combating misinformation on the web with ai and blockchain. pages 435–442, 2022.

[96] M. Shafay, R. W. Ahmad, K. Salah, I. Yaqoob, R. Jayaraman, and M. Omar. Blockchain for deep learning: review and open challenges. *Cluster Computing*, 26(1):197–221, 2023.

[97] Z. Shah, I. Ullah, H. Li, A. Levula, and K. Khurshid. Blockchain based solutions to mitigate distributed denial of service (ddos) attacks in the internet of things (iot): A survey. 22, 2022.

[98] M. Sharma, M. Tong, T. Korbak, D. Duvenaud, A. Askell, S. R. Bowman, N. Cheng, E. Durmus, Z. Hatfield-Dodds, S. R. Johnston, S. Kravec, T. Maxwell, S. McCandlish, K. Ndousse, O. Rausch, N. Schiefer, D. Yan, M. Zhang, and E. Perez. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023.

[99] E. Shayegani, M. A. A. Mamun, Y. Fu, P. Zaree, Y. Dong, and N. Abu-Ghazaleh. Survey of vulnerabilities in large language models revealed by adversarial attacks. *arXiv preprint arXiv:2310.10844*, 2023.

[100] M. Shen, H. Wang, B. Zhang, L. Zhu, K. Xu, Q. Li, and X. Du. Exploiting unintended property leakage in blockchain-assisted federated learning for intelligent edge computing. *IEEE Internet of Things Journal*, 8(4):2265–2275, 2021.

[101] T. Shen, R. Jin, Y. Huang, C. Liu, W. Dong, Z. Guo, X. Wu, Y. Liu, and D. Xiong. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*, 2023.

[102] S. Singh. Enhancing privacy and security in large-language models: A zero-knowledge proof approach. *International Conference on Cyber Warfare and Security*, 19(1):574–582, 2024.

[103] S. Singh, P. K. Sharma, B. Yoon, M. Shojafar, G. H. Cho, and I.-H. Ra. Convergence of blockchain and artificial intelligence in IoT network for the sustainable smart city. *Sustainable Cities and Society*, 63, 2020.

[104] H. Song, Z. Qu, and Y. Wei. Advancing blockchain scalability: An introduction to layer 1 and layer 2 solutions. *arXiv preprint arXiv:2406.13855*, 2024.

[105] H. Song, Y. Wei, Z. Qu, and W. Wang. Unveiling decentralization: A comprehensive review of technologies, comparison, challenges in bitcoin, ethereum, and solana blockchain. *arXiv preprint arXiv:2404.04841*, 2024.

[106] T. South, G. Zuskind, R. Mahari, and T. Hardjono. Secure community transformers: Private pooled data for llms. 2023.

[107] L. Sun, Y. Huang, et al. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*, 2024.

[108] X. Sun, F. R. Yu, P. Zhang, Z. Sun, W. Xie, and X. Peng. A survey on zero-knowledge proof in blockchain. *IEEE Network*, 35(4), 2021.

[109] H. Taherdoost. Blockchain technology and artificial intelligence together: A critical review on applications. *Applied Sciences*, 12(24):12948, 2022.

[110] S. Tedeschi, F. Friedrich, P. Schramowski, K. Kersting, R. Navigli, H. Nguyen, and B. Li. Alert: A comprehensive benchmark for assessing large language models' safety through red teaming. *arXiv preprint arXiv:2404.08676*, 2024.

[111] T. Teubner, C. M. Flath, C. Weinhardt, W. van der Aalst, and O. Hinz. Welcome to the era of chatgpt et al. *Business & Information Systems Engineering*, 65, 2023.

[112] C. Tonkin. 'ChatGPT, help me make a bomb', 2023.

[113] I. Ullah, N. Hassan, S. S. Gill, B. Suleiman, T. A. Ahanger, Z. Shah, J. Qadir, and S. S. Kanhere. Privacy preserving large language models: Chatgpt case study based vision and framework. *arXiv preprint arXiv:2310.12523*, 2023.

[114] Y. Wan, G. Pu, J. Sun, A. Garimella, K.-W. Chang, and N. Peng. "kelly is a warm person, joseph is a role model": Gender biases in llm-generated reference letters. *arXiv preprint arXiv:2310.09219*, 2023.

[115] B. Wang, W. Chen, H. Pei, C. Xie, M. Kang, C. Zhang, C. Xu, Z. Xiong, R. Dutta, R. Schaeffer, S. T. Truong, S. Arora, M. Mazeika, D. Hendrycks, Z. Lin, Y. Cheng, S. Koyejo, D. Song, and B. Li. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. *NeurIPS*, 2023.

[116] Q. Wang, Y. Guo, X. Wang, T. Ji, L. Yu, and P. Li. Ai at the edge: Blockchain-empowered secure multiparty learning with heterogeneous models. *IEEE Internet of Things Journal*, 7(10):9600–9610, 2020.

[117] L. Weidinger, M. Rauh, N. Marchal, A. Manzini, L. A. Hendricks, J. Mateos-Garcia, S. Bergman, J. Kay, C. Griffin, B. Bariach, I. Gabriel, V. Rieser, and W. Isaac. Sociotechnical safety evaluation of generative ai systems. *arXiv preprint arXiv:2310.11986*, 2023.

[118] S. Wellington. Basedai: A decentralized p2p network for zero knowledge large language models (zk-llms). *arXiv preprint arXiv:2403.01008*, 2024.

[119] C. Weng, K. Yang, X. Xie, J. Katz, and X. Wang. Mystique: Efficient conversions for {Zero-Knowledge} proofs with applications to machine learning. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 501–518, 2021.

[120] J. Weng, J. WENG, M. Li, Y. Zhang, J. ZHANG, and W. LUO. Auditable privacy protection deep learning platform construction method based on block chain incentive mechanism, 2023. US Patent 11,836,616.

[121] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo. Deepchain: Auditable and privacy-preserving

deep learning with blockchain-based incentive. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2438–2455, 2021.

[122] K. Werder, B. Ramesh, and R. S. Zhang. Establishing data provenance for responsible artificial intelligence systems. *ACM Transactions on Management Information Systems*, (2):1–23, 2022.

[123] A. Winograd. Loose-lipped large language models spill your secrets: The privacy implications of large language models notes. *Harvard Journal of Law & Technology (Harvard JOLT)*, (2), 2022.

[124] L. Witt, A. T. Fortes, K. Toyoda, W. Samek, and D. Li. Blockchain and artificial intelligence: Synergies and conflicts. *arXiv preprint arXiv:2405.13462*, 2024.

[125] C. F. Xavier. Pipemsm: Hardware acceleration for multi-scalar multiplication. *Cryptology ePrint Archive*, 2022.

[126] J. Xue, M. Zheng, Y. Hu, F. Liu, X. Chen, and Q. Lou. Badrag: Identifying vulnerabilities in retrieval augmented generation of large language models. *arXiv preprint arXiv:2406.00083*, 2024.

[127] B. Yan, K. Li, M. Xu, Y. Dong, Y. Zhang, Z. Ren, and X. Cheng. On protecting the data privacy of large language models (llms): A survey. *arXiv preprint arXiv:2403.05156*, 2024.

[128] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, B. Yin, and X. Hu. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32, 2023.

[129] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211, 2024.

[130] A. Yazdinejad, R. M. Parizi, G. Srivastava, and A. Dehghantanha. Making sense of blockchain for ai deepfakes technology. pages 1–6, 2020.

[131] H. Ye, T. Liu, A. Zhang, W. Hua, and W. Jia. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*, 2023.

[132] S. Zeng, J. Zhang, P. He, Y. Xing, Y. Liu, H. Xu, J. Ren, S. Wang, D. Yin, Y. Chang, et al. The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag). *arXiv preprint arXiv:2402.16893*, 2024.

[133] J. Zhang, T. Xie, Y. Zhang, and D. Song. Transparent polynomial delegation and its applications to zero knowledge proof. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 859–876. IEEE, 2020.

[134] R. Zhang, H. Du, Y. Liu, D. Niyato, J. Kang, S. Sun, X. Shen, and H. V. Poor. Interactive ai with retrieval-augmented generation for next generation networking. *IEEE Network*, 2024.

[135] R. Zhang, R. Xue, and L. Liu. Security and privacy on blockchain. 52(3), 2020.

[136] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, L. Wang, A. T. Luu, W. Bi, F. Shi, and S. Shi. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.

[137] Z. Zhang, L. Lei, L. Wu, R. Sun, Y. Huang, C. Long, X. Liu, X. Lei, J. Tang, and M. Huang. Safetybench: Evaluating the safety of large language models with multiple choice questions. *arXiv preprint arXiv:2309.07045*, 2023.

[138] Z. Zhang, Y. Rao, H. Xiao, X. Xiao, and Y. Yang. Proof of quality: A costless paradigm for trustless generative ai model inference on blockchains. *arXiv preprint arXiv:2405.17934*, 2024.

[139] Z. Zhang, J. Yang, P. Ke, F. Mi, H. Wang, and M. Huang. Defending large language models against jailbreaking attacks through goal prioritization. *arXiv preprint arXiv:2311.09096*, 2024.

[140] S. Zhao, J. Wen, A. Luu, J. Zhao, and J. Fu. Prompt as triggers for backdoor attack: Examining the vulnerability in language models. 2023.

[141] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[142] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, L. Lyu, and Y. Liu. Privacy-preserving blockchain-based federated learning for iot devices. *IEEE Internet of Things Journal*, 8(3):1817–1829, 2021.

[143] Y. Zhao, J. Zhao, J. Kang, Z. Zhang, D. Niyato, S. Shi, and K.-Y. Lam. A blockchain-based approach for saving and tracking differential-privacy cost. *IEEE Internet of Things Journal*, 8(11):8865–8882, 2021.

[144] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

[145] Z. Zheng, K. Ning, Y. Wang, J. Zhang, D. Zheng, M. Ye, and J. Chen. A survey of large language models for code: Evolution, benchmarking, and future trends. *arXiv preprint arXiv:2311.10372*, 2023.

[146] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran. An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*, 105:475–491, 2020.

[147] X. Zhu, H. Li, and Y. Yu. Blockchain-based privacy preserving deep learning. pages 370–383, 2019.

[148] J. Zhuang and C. Kennington. Understanding survey paper taxonomy about large language models via graph representation learning. *arXiv preprint arXiv:2402.10409*, 2024.

[149] W. Zou, R. Geng, B. Wang, and J. Jia. Poisonedrag: Knowledge poisoning attacks to retrieval-augmented generation of large language models. *arXiv preprint arXiv:2402.07867*, 2024.

[150] X. Zuo, M. Wang, T. Zhu, L. Zhang, D. Ye, S. Yu, and W. Zhou. Federated trustchain: Blockchain-enhanced llm training and unlearning. *arXiv preprint arXiv:2406.04076*, 2024.

[151] X. Zuo, M. Wang, T. Zhu, L. Zhang, S. Yu, and W. Zhou. Federated learning with blockchain-enhanced machine unlearning: A trustworthy approach. *arXiv preprint arXiv:2405.20776*, 2024.

Authorship Attribution in the Era of LLMs: Problems, Methodologies, and Challenges

Baixiang Huang
Illinois Institute of Technology
Chicago, IL
bhuang15@hawk.iit.edu

Canyu Chen
Illinois Institute of Technology
Chicago, IL
cchen151@hawk.iit.edu

Kai Shu
Emory University
Atlanta, GA
kai.shu@emory.edu

ABSTRACT

Accurate attribution of authorship is crucial for maintaining the integrity of digital content, improving forensic investigations, and mitigating the risks of misinformation and plagiarism. Addressing the imperative need for proper authorship attribution is essential to uphold the credibility and accountability of authentic authorship. The rapid advancements of Large Language Models (LLMs) have blurred the lines between human and machine authorship, posing significant challenges for traditional methods. We present a comprehensive literature review that examines the latest research on authorship attribution in the era of LLMs. This survey systematically explores the landscape of this field by categorizing four representative problems: (1) Human-written Text Attribution; (2) LLM-generated Text Detection; (3) LLM-generated Text Attribution; and (4) Human-LLM Co-authored Text Attribution. We also discuss the challenges related to ensuring the generalization and explainability of authorship attribution methods. Generalization requires the ability to generalize across various domains, while explainability emphasizes providing transparent and understandable insights into the decisions made by these models. By evaluating the strengths and limitations of existing methods and benchmarks, we identify key open problems and future research directions in this field. This literature review serves a roadmap for researchers and practitioners interested in understanding the state of the art in this rapidly evolving field. Additional resources and a curated list of papers are available and regularly updated at https://llm-authorship.github.io/.

1. INTRODUCTION

Authorship Attribution (AA) is the process of determining the author of a particular piece of writing and has significant real-world applications across various domains. In forensic investigations, authorship attribution plays a crucial role in solving murder cases disguised as suicides [Chaski, 2005; Grant, 2020], tracking terrorist threats [Winter, 2019; Cafiero and Camps, 2023], and aiding general criminal investigations [Koppel et al., 2008; Argamon, 2018; Belvisi et al., 2020]. In the digital realm, authorship attribution helps safeguard the integrity of content by preventing deceptive social media activities [Hazell, 2023], detecting account compromises [Barbon et al., 2017], and linking user profiles across various social networks [Shu et al., 2017; Sinnott and Wang, 2021]. Additionally, authorship attribution techniques are instrumental in combating misinformation [Shu et al., 2020; Chen and Shu, 2024a; Chen et al., 2022b; Hanley and Durumeric, 2024; Stiff and Johansson, 2022], protecting intellectual property rights [Meyer zu Eissen et al., 2007; Stamatatos and Koppel, 2011], and identifying fraudulent activities [Ott et al., 2011; Afroz et al., 2012].

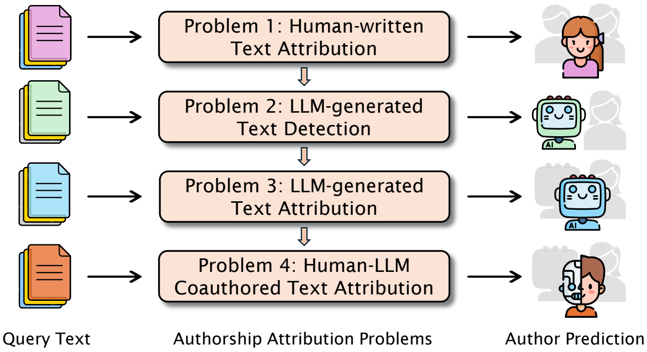

Problem 1: Human-written Text Attribution
Problem 2: LLM-generated Text Detection
Problem 3: LLM-generated Text Attribution
Problem 4: Human-LLM Coauthored Text Attribution
AI
Query Text
Authorship Attribution Problems
Author Prediction

Figure 1: Representative Problems in Authorship Attribution: (1) Human-written Text Attribution, which involves attributing an unknown text to its human authors; (2) LLM-generated Text Detection, which focuses on detecting whether a text has been generated by LLMs; (3) LLM-generated Text Attribution, aimed at identifying the specific LLM or human responsible for a given text; (4) Human-LLM Co-authored Text Attribution, which classifies a text as human-written, LLM-generated, or a combination of both. *The categorization of these problems becomes increasingly complex, as indicated by the arrow, balancing complexity with practicality.*

The development of large language models (LLMs) has revolutionized text generation, offering numerous benefits but also raising significant concerns about text authenticity and originality [Brown et al., 2020; Goldstein et al., 2023]. The advent of LLMs has complicated authorship attribution, making it increasingly difficult to distinguish between LLM-generated texts and human-written texts [Clark et al., 2021; Sadasivan et al., 2023]. Identifying LLM-generated texts is challenging even for human experts, let alone traditional authorship attribution methods [Liu et al., 2023a; Gao et al., 2022]. This inability to distinguish between human and machine-generated content undermines the integrity of authorship, complicates legal and ethical responsibilities, and threatens the credibility of digital content and the safety of online space [Solaiman et al., 2023; Vidgen et al., 2024].

Over the past few decades, authorship attribution has experienced significant advancements due to the development of natural language analysis tools and innovative methods for text representation learning. Traditionally, authorship attribution relied on stylometry, which analyzes an individual's unique writing style through feature engineering to capture linguistic characteristics [Lagutina et al., 2019]. The emergence of machine learning algorithms capable of handling high-dimensional data has enabled the creation of more expressive representations. In recent years, there has been a shift towards extracting text embeddings using pre-trained language models [Fabien et al., 2020]. These approaches, while offering higher performance, often sacrifice explainability for accuracy [Rivera-Soto et al., 2021]. More recently, researchers have begun to use LLMs to extract features in conjunction with machine learning classifiers or to conduct end-to-end reasoning for authorship attribution [Patel et al., 2023; Huang et al., 2024].

Rapid advancements in LLM have significantly improved text generation, producing outputs that rival human writing in fluency and coherence. This progress underscores the imperative need to distinguish between human-written text, LLM-generated text, or a combination of both. As illustrated in Figure 1, authorship attribution can be systematically categorized into four representative problems: attributing unknown texts to human authors, detecting LLM-generated texts, identifying the specific LLM or human responsible for a text, and classifying texts as human, machine, or human-LLM co-authored. Each task presents unique challenges that necessitate corresponding solutions. Researchers continually adapt and refine attribution methods, transitioning from human-authored texts to LLM-generated content, and navigating the complex interweaving in human-LLM co-authored works. As detection methods advance, adversarial attacks also evolve to bypass these measures, creating a continuous cycle of challenge and response in the quest to distinguish and disguise authorship [Dugan et al., 2024]. Addressing these challenges will pave the way for more robust and reliable authorship attribution techniques.

Authorship attribution for both human and LLM-generated texts can be framed as either binary or multi-class classification. The LLM-generated text detection task simplifies the attribution problem by classifying each text as either originating from humans or LLMs [Jawahar et al., 2020a; Mitchell et al., 2023; Pu et al., 2023a; Sadasivan et al., 2023]. The majority of previous research on automatic detection of machine-generated text has focused on binary classification [Jawahar et al., 2020b; Mitchell et al., 2023]. In the more challenging multi-class task, the goal is not only to differentiate between human and LLM-generated text but also to classify the text according to its specific source of generative models [Uchendu et al., 2021; Li et al., 2023b]. Differences in LLM architectures, training methods, and generation techniques can influence the style of generated texts [Munir et al., 2021]. In the more complex human-LLM co-authoring problem, the goal is to distinguish texts authored by humans, LLMs, or combinations of both. Such nuanced detection provides deeper insight into the provenance of the text and is crucial for applications requiring detailed source attribution. Neural network-based detectors generally outperform metric-based methods in both human authorship attribution and LLM-generated text detection problems [He et al., 2023; Zhang et al., 2024]. However, these neural network approaches often offer less explainability compared to their metric-based counterparts.

This review serves as a valuable resource, comprehensively summarizing the existing literature and highlighting the challenges and opportunities introduced by LLMs. We provide in-depth analysis of methodologies in this evolving field. The main contributions of this paper are as follows:

- We provide a timely overview to discuss the challenges and opportunities presented by LLMs in the field of authorship attribution. By systematically categorizing authorship attribution into four representative problems and balancing problem complexity with practicality, we reveal insights into the evolving field of authorship attribution in the era of LLMs.
- We offer a comprehensive comparison of state-of-the-art methodologies, datasets, benchmarks, and commercial tools used in authorship attribution. This analysis not only improves the understanding of authorship attribution but also provides a valuable resource for researchers and practitioners to use as guidelines for approaching this direction.
- We discuss open issues and provide future directions by considering crucial aspects such as generalization, explainability, and interdisciplinary perspectives. We also discuss the broader implications of authorship attribution in real-world applications. This holistic approach ensures that authorship attribution not only yields accurate results but also provides insights that are explainable and socially relevant.

The remainder of this survey is organized as follows. Section 2 explores the attribution of human authorship, beginning with a definition of the problem, followed by an overview of various methodologies and a discussion of the associated challenges. In Section 3, we discuss LLM-generated text detection. In Section 4, we explore the attribution of LLM-generated text. Section 5 covers the attribution of human-LLM co-authored texts. Section 6 discusses resources and evaluation metrics, offering a comparison of benchmarks and datasets. Section 7 highlights opportunities and future directions. In Section 8, we discuss ethical and privacy concerns. Finally, we conclude this survey in Section 9.

2. HUMAN AUTHORSHIP ATTRIBUTION

This section explores the authorship attribution of human-written texts, discussing a range of methodologies such as stylometry, machine learning, pre-trained language models, and LLM-based approaches. It also discusses challenges such as limited data, evolving writing styles, and interpretability.

2.1 Problem Definition

Authorship attribution aims to identify the author of an unknown text from a set of known authors. This can be formulated as an open-class problem, where the true author might not be among the known authors, or a closed-class problem, where the true author is included in a finite set of authors [Stolerman et al., 2014; Andrews and Bishop, 2019]. Authorship attribution methods are typically divided into classification-based methods for a small set of candidate authors and similarity-based ranking methods for larger numbers of authors [Rivera-Soto et al., 2021; Huertas-Tato et al., 2022]. These techniques can also be adapted to related problems, such as authorship verification and profiling. Authorship verification determines whether a piece of writing was written by a specific individual [Stamatatos, 2016], while profiling infers characteristics such as age or gender from the author's writing style [Argamon et al., 2009].

2.2 Methodologies

We explore the evolution of methods used to analyze human-written text, beginning with stylometry. Over time, the focus has shifted to the use of machine learning. Recently, the integration of LLMs marks further advancements in authorship attribution.

2.2.1 *Stylometry Methods*

Stylometry, the quantitative analysis of writing style, has evolved from its initial reliance on human expertise [Mosteller and Wallace, 1963] to computational methods [Neal et al., 2017; Lagutina et al., 2019]. This discipline utilizes a variety of linguistic features to determine the authorship [Holmes, 1994; Lagutina et al., 2019], positing that each author's unique style can be captured through quantifiable characteristics [Argamon et al., 2009]. Key stylometric features include character and word frequencies [Sharma et al., 2018], parts of speech [Sundararajan and Woodard, 2018], punctuation, topics [Seroussi et al., 2014; Potha and Stamatatos, 2019; Halvani and Graner, 2021], and vocabulary richness. Important features can be categorized into the following types: lexical, syntactic, semantic, structural, and content-specific [Rudman, 1997]. Lexical features involve word choice and frequency; syntactic features pertain to sentence structure and grammar; semantic features explore the meaning and context of words; structural features relate to text organization; and content-specific features emphasize domain-specific terms [Bozkurt et al., 2007; Seroussi et al., 2014].

2.2.2 *Machine Learning Methods*

Machine learning approaches integrate stylometric features with classifiers such as logistic regression [Aborisade and Anwar, 2018; Madigan et al., 2005a], Bayesian multinomial regression [Grant, 2007; Argamon et al., 2009], and support vector machines (SVM) [Bacciu et al., 2019]. Before widespread adoption of transformer-based models, multi-headed Recurrent Neural Networks (RNNs) [Bagnall, 2015], and Long Short-Term Memory (LSTMs) were utilized at both sentence and article levels [Qian et al., 2017]. Convolutional Neural Networks (CNNs) were also applied at various levels, including characters, words, and N-grams [Ruder et al., 2016; Shrestha et al., 2017a,b]. Moreover, syntax-augmented CNN models [Zhang et al., 2018], Convolutional Siamese Networks [Saedi and Dras, 2021], and attention-based Siamese Networks [Boenninghoff et al., 2019a] were explored. Additionally, combinations of convolutions and transformers have been employed to learn embeddings for comparison tasks [Andrews and Bishop, 2019].

2.2.3 *Pre-trained Language Models*

Pre-trained language models (PTMs), especially BERT-based architectures [Devlin et al., 2019] such as BERT [Ippolito et al., 2020; Fabien et al., 2020; Manolache et al., 2021], Sentence-BERT [Schlicht and de Paula, 2021; Rivera-Soto et al., 2021], and RoBERTa [Huertas-Tato et al., 2022], have proven effective for learning authorship representation. These methods do not require hand-crafted features but require substantial training time and domain-specific labeled data, struggling with cross-domain generalization and explainability. Contrastive learning [Khosla et al., 2020] is often used with pre-trained language models to enhance stylistic representation by maximizing similarity between texts written by the same authors and minimizing it with texts from different authors [Huertas-Tato et al., 2022].

Barlas and Stamatatos [2020a] found that BERT performed well with large vocabularies, outperforming multi-headed RNNs. Fabien et al. [2020] fine-tuned a BERT model, showing that including additional stylometric and hybrid features in an ensemble model can improve attribution performance. Rivera-Soto et al. [2021] concluded that topic diversity and dataset size are crucial for effective cross-domain transfer. Adaptation through style transfer has not resolved cross-domain issues [Boenninghoff et al., 2019b; Wegmann et al., 2022]. Techniques like slanted triangular learning rates and gradual unfreezing can be used to avoid catastrophic forgetting during fine-tuning [Howard and Ruder, 2018].

2.2.4 *LLM-based Methods*

Despite advances in LLMs, their potential for authorship attribution remains underexplored. The natural language understanding capability allows LLMs to recognize nuances, styles, and patterns in language, which are crucial for distinguishing between authors. Authorship attribution is a complex reasoning task, and LLMs possess significant capabilities in reasoning and problem-solving, particularly in zero-shot learning within resource-limited domains [Kojima et al., 2022]. They assist in feature extraction by identifying syntactic patterns, lexical choices, and grammatical structures essential for authorship attribution. Traditionally, LLMs have been employed mainly for auxiliary tasks such as feature extraction and data annotation [Patel et al., 2023]. Notable examples include the use of GPT-3 for data annotation [Brown et al., 2020] and a T5 encoder for learning authorship signatures. Beyond feature extraction, LLMs possess the ability to identify the author of unknown text based on nuanced linguistic features, making it possible to conduct end-to-end authorship attribution [Huang et al., 2024].

The incorporation of LLMs in authorship attribution addresses several limitations of traditional methods. Unlike BERT-based models, which require computationally expensive fine-tuning and large amounts of domain-specific data for optimal performance, LLMs can generalize across various domains without fine-tuning, thereby mitigating issues related to domain specificity [Barlas and Stamatatos, 2020a]. LLMs are also effective with shorter texts, reducing the necessity for long inputs to derive meaningful representations [Eder, 2015]. Another key advantage of LLM-based approaches is their ability to provide natural language explanations for their predictions, enhancing transparency compared to hidden text embeddings [Huang et al., 2024]. This versatility marks a step forward in overcoming challenges related to data, domain specificity, text length requirements, and explainability faced by earlier methods.

2.3 Open Challenges

Human authors exhibit a diverse range of writing styles influenced by genre, topic, context, and temporal changes. This variability complicates authorship attribution, necessitating the identification of consistent and unique stylistic markers. Further complexity arises from the presence of noise due to the varying size and language of documents, requiring algorithms to identify linguistic nuances. Short or low-quality texts, such as social media posts, are often unreliable, complicating accurate attribution [Eder, 2015; Theophilo et al., 2021]. Collaborative texts, which blend multiple writing styles, further mask individual contributions and obscure distinct authorial signals [Dauber et al., 2017].

Traditional stylometric methods rely on human expertise and manually crafted features, whereas deep learning methods demand significant computational resources and extensive labeled data, with the risk of catastrophic forgetting [Ramasesh et al., 2022]. Authorship attribution using LLMs also faces several challenges: their effectiveness decreases with an increasing number of candidate authors due to context length constraints [Huang et al., 2024], and they can perpetuate biases from training data, resulting in inaccurate attributions for texts from marginalized groups and languages [Liang et al., 2023]. Additionally, LLMs can be misused to generate content that conceals true authorship by mimicking others or using LLMs to alter their work.

3. LLM-GENERATED TEXT DETECTION

LLMs excel at generating fluent and coherent text, which raises concerns about the authenticity and originality of the resulting work. Detecting LLM-generated text is crucial for several applications, including combating misinformation on social media [Gambini et al., 2022; Stiff and Johansson, 2022; Chen and Shu, 2024b], identifying spam [Jindal and Liu, 2008], preventing phishing attacks [Hazell, 2023], identifying fake reviews [Salminen et al., 2022], and detecting machine-generated scientific papers [Rodriguez et al., 2022a; Liu et al., 2023a]. As a result, the detection of LLM-generated text has garnered significant attention [Kumarage and Liu, 2023; Tang et al., 2023a; Wu et al., 2023b; Yang et al., 2023c].

3.1 Problem Definition

LLM-generated texts are included within the scope of machine-generated texts[1]. Machine-generated texts encompass any text produced by automated systems, including simpler language models or rule-based systems [Uchendu et al., 2021]. This paper focuses specifically on LLM-generated texts. The task of detecting LLM-generated text involves distinguishing text created by LLMs from that written by humans. Typically, this task is approached as a binary classification problem [Zellers et al., 2019; Solaiman et al., 2019; Jawahar et al., 2020b; Fagni et al., 2021; Mitchell et al., 2023].

3.2 Methodologies

The evaluation of the quality of machine-generated excerpts has traditionally relied on human judgement, which is considered the gold standard for open-domain generation systems [van der Lee et al., 2019; Gehrmann et al., 2019a]. However, distinguishing between LLM-generated and human-written texts poses significant challenges for humans [Dugan et al., 2023]. For example, untrained human reviewers are often unable to distinguish GPT-3-generated text from human-written text, identifying it correctly only at a rate consistent with random chance [Clark et al., 2021]. Liu et al. [2023a] found that even experienced faculty and researchers could only achieve about a 50% success rate in identifying GPT-generated academic writings. In contrast, detection algorithms frequently outperform the human in this task [Ippolito et al., 2020].

Chakraborty et al. [2023a] employed theoretical analysis to argue that detecting LLM-generated text is nearly always feasible with the collection of multiple samples, and they established precise sample complexity bounds for this detection. However, existing detectors and models for LLM-generated text are not yet fully reliable [Sadasivan et al., 2023; Wang et al., 2023; Dugan et al., 2024]. Sadasivan et al. [2023] provided theoretical insights indicating that the detection problem is becoming increasingly difficult.

LLM-generated text detectors can be categorized into metric-based and model-based methods [He et al., 2023], which are further divided into feature-based, neural network-based, zero-shot-based, and watermark-based methods. These detectors are also classified as white-box or black-box, depending on their access to the LLM weights [Tang et al., 2023b; Yang et al., 2023c]. Watermarking-based methods typically fall under the white-box detection category, while proprietary models are restricted to black-box methods.

3.2.1 *Featured-based Method*

LLM-generated texts are typically less emotional and more objective than human-written texts [Guo et al., 2023]. Human-authored texts are generally more coherent, while LLM-generated texts tend to repeat terms within a paragraph [Dugan et al., 2023]. Similarly to the problem of attribution of human authorship, linguistic characteristics such as phrasal verbs, co-reference, part-of-speech (POS) tags, and named entity (NE) tags are also useful in distinguishing LLM-generated text [Nguyen-Son et al., 2017; See et al., 2019; Fröhling and Zubiaga, 2021]. Feature-based methods are more explainable but have drawbacks, such as poor generalizability of certain features across different domains and sampling methods.

3.2.2 *Neural Network-Based Detectors*

Neural network-based detectors, particularly those utilizing BERT, have proven effective in distinguishing between human-written texts and those generated by GPT-2 [Ippolito et al., 2020; Liu et al., 2019]. Solaiman et al. [2019] fine-tuned the RoBERTa model with a dataset of GPT-2 outputs in open domain settings. Similarly, Guo et al. [2023] fine-tuned RoBERTa to detect ChatGPT-generated text. [Zhan et al., 2023] developed G3Detector by fine-tuning RoBERTa-large for the same purpose. Additionally, Chen et al. [2023a] introduced GPT-Sentinel, training both RoBERTa and T5 [Raffel et al., 2020] on their OpenGPTText dataset. In a different approach, Hu et al. [2023] created RADAR, which fine-tunes

[1] Machine-generated texts is also referred to as machine-authored, AI-generated, neural-generated, deepfake text, neural text, or synthetic text.

Vicuna 7B [Chiang et al., 2023] in a generative adversarial setting along with a paraphrase model. These efforts highlight ongoing advances in the detection of LLM-generated content using BERT-based models.

These detectors require retraining when encountering text generated by new LLMs to ensure reliable detection [Mitchell et al., 2023; Chakraborty et al., 2023b]. On the other hand, neural network-based detectors are vulnerable to adversarial and poisoning attacks [Goodfellow et al., 2015; Wang et al., 2022; Pu et al., 2023a]. These detectors also face limitations such as overfitting to training data [Uchendu et al., 2020]. The generalization ability of these detectors is critical, as they have been trained in various family models and tested in unseen models [Pu et al., 2023b; Bhattacharjee et al., 2023]. Surrogate models, which are often small language models, are also applied to train classifiers [Verma et al., 2023; Mireshghallah et al., 2023].

3.2.3 *Zero-Shot Detectors*

Zero-shot detection methods are generally statistics-based, enabling the detection of LLM-generated text without additional training [Su et al., 2023b]. Various statistical measures have been employed, including entropy [Lavergne et al., 2008; Gehrmann et al., 2019a], perplexity [Beresneva, 2016; Hans et al., 2024], average log-probability score [Solaiman et al., 2019], fluency [Holtzman et al., 2020], and Zipf's word frequency law [Zipf, 2016; Piantadosi, 2014][2]. Additional methods leverage n-grams [Yang et al., 2023b], Uniform Information Density (UID) [Venkatraman et al., 2023], log rank information [Su et al., 2023c], and various linguistic features such as part-of-speech determiners, conjunctions, auxiliary relations, vocabulary, and emotional tone [Joulin et al., 2017; Tang et al., 2023b; Gehrmann et al., 2019b].

Various zero-shot detection methods provide distinct strategies to enhance both detection accuracy and efficiency. Gallé et al. [2021] introduced an unsupervised method that identifies the over-appearance of repeated higher-order n-grams, distinguishing them from human-generated text. DetectGPT [Mitchell et al., 2023] is based on the observation that LLM-generated passages often fall into regions of negative curvature in log probability. [Bao et al., 2023] enhanced this approach and proposed Fast-DetectGPT, which increases efficiency by using conditional probability curvature. Additionally, some methods leverage LLMs themselves for text classification [Zellers et al., 2019; Solaiman et al., 2019]. Different decoding methods are often applied to generate more diverse and less repetitive text, although these can also lead to hallucinations and less verifiable content [Shakeel and Jain, 2021; Guo et al., 2023]. Fact-checking methods can mitigate these issues [Zhong et al., 2020; Schuster et al., 2020]. Additionally, Krishna et al. [2024a] developed a detector that uses information retrieval to store LLM output in a database and search for semantically similar content to identify LLM-generated text, although this method raises privacy concerns regarding the storage of user conversations.

3.2.4 *Watermarking*

Watermarking involves embedding specific patterns in text, making them imperceptible to humans but detectable through specialized methods [Topkara et al., 2005; Meral et al., 2009; Kirchenbauer et al., 2023; Zhao et al., 2023]. By imprinting distinct patterns, watermarking enables the identification of LLM-generated text. Various methods include parsed syntactic tree structures [Atallah et al., 2001; Topkara et al., 2005], synonym tables [Jalil and Mirza, 2009], adversarial watermarking [Abdelnabi and Fritz, 2021], and context-aware lexical substitution [Yang et al., 2022]. One notable approach is soft watermarking, proposed by Kirchenbauer et al. [2023], which partitions tokens into "green" and "red" lists to create patterns. A watermarked LLM samples tokens from the green list with high probability, determined by a pseudo-random generator seeded by its prefix token. The watermarking detector classifies passages with a high frequency of tokens from the green list as LLM-generated.

Other studies have improved the robustness, efficiency, and stealthiness of watermarking methods [Hou et al., 2023; Wu et al., 2023a; Zhao et al., 2023]. However, there is a trade-off between watermark effectiveness and text quality, as more reliable watermarks require more extensive text modifications [Sadasivan et al., 2023]. Additionally, watermarking presents challenges for proprietary LLMs and third-party applications due to the necessity of accessing the language model logits [Kirchenbauer et al., 2023]. Watermark-based detection methodologies are also vulnerable to paraphrasing attacks [Sadasivan et al., 2023; Krishna et al., 2024a].

3.3 Open Challenges

Detecting LLM-generated texts is challenging due to their versatile styles and contextual awareness. LLMs can emulate human writing so closely that they pose significant challenges to traditional stylometric techniques. They can incorporate complex narrative structures and varied vocabularies, making it difficult to distinguish between human and LLM-generated texts. The rapid evolution of LLMs further complicates detection, as newer versions exhibit different stylistic characteristics, making detection models quickly obsolete [Chakraborty et al., 2023b]. LLM-generated text detectors often struggle to generalize to unseen domains encountered during training [Pu et al., 2023b; Rodriguez et al., 2022b; Li et al., 2023a] and tend to perform better on LLMs they were specifically trained on [Pu et al., 2023b; Chakraborty et al., 2023b; Li et al., 2023a].

Existing detectors also lack robustness to various factors, such as alternative decoding strategies [Ippolito et al., 2020], input sequence length [Solaiman et al., 2019], different prompts [Kumarage et al., 2023; Lu et al., 2023], and repetition penalties [Fishchuk and Braun, 2023]. Additionally, detectors are vulnerable to adversarial attacks, including homoglyph attacks [Gagiano et al., 2021; Macko et al., 2024], whitespace insertion [Cai and Cui, 2023], syntactic perturbations [Bhat and Parthasarathy, 2020a], synonym replacement [Kulkarni et al., 2023], and paraphrasing [Krishna et al., 2024b; Shi et al., 2024; Becker et al., 2023].

4. LLM-GENERATED TEXT ATTRIBUTION

Identifying whether a piece of text is generated by a specific LLM or a human is crucial. This distinction helps to trace the origin of the text to ensure accountability, enhance transparency, and uphold ethical standards in information dissemination. If the content is harmful, misleading, or illegal,

[2] The frequency of a word decreases as its rank in a frequency-ordered list increases

pinpointing the exact responsible LLM is essential to address ethical concerns and fulfill legal obligations. LLM-generated text attribution builds upon techniques for LLM-generated text detection. Variations in model architecture (such as the number of layers and parameters), training methods (including pre-training and fine-tuning), and generation techniques (such as sampling parameters) influence all the characteristics of the generated texts [Munir et al., 2021].

4.1 Problem Definition

This attribution task extends beyond binary classification to handle multiple classes, increasing the complexity of LLM-generated text detection. The primary goal is to determine whether a given piece of text was created by a specific human or by one of several LLMs [Uchendu et al., 2020; Venkatraman et al., 2023; Chen et al., 2023b; He et al., 2023; Soto et al., 2024]. A sub-problem is to attribute the text solely to LLMs, also known as model sourcing [Yang et al., 2023a] or origin tracing [Li et al., 2023b].

4.2 Methodologies

Attributing texts to LLMs versus human writers involves recognizing inherent differences in their text generation capabilities. LLMs typically exhibit less diversity in word usage compared to humans [Ippolito et al., 2020; Dugan et al., 2023]. LLMs can mimic a range of styles and tones, often masking their underlying characteristics. This ability to adapt makes attribution challenging, especially as LLMs rapidly evolve and their outputs change significantly over time [Guo et al., 2023]. Additionally, LLMs may inadvertently reproduce snippets of their training data.

To simplify the classification process, it is common practice to group different human writers into a single category because humans exhibit a broader spectrum of writing styles and proficiency levels compared to machines [Uchendu et al., 2021]. For example, classifications might include comparisons such as Human vs. ChatGPT, or Human vs. LLama [Uchendu et al., 2021; He et al., 2023]. Some studies have formulated a seven-class classification that includes one human class and six LLM classes [He et al., 2023; Wang et al., 2024]. Other approaches consider multiple human classes, albeit with a limited number. For instance, a 10-class classification might include seven human classes and three LLM classes [Tripto et al., 2023]. This multiclass classification is often converted into a one-vs-rest classification for each label. Transformer-based models, such as BERT and RoBERTa, are fine-tuned on datasets containing both human-written and LLM-generated texts to conduct the attribution.

4.3 Open Challenges

Attributing texts generated by LLMs to specific humans or models presents a multi-class classification challenge. Variations in training data, model architecture, and fine-tuning processes contribute to the distinctive outputs of different LLMs, though these differences are often subtle and difficult to detect [Uchendu et al., 2021]. Effective identification requires sophisticated methods to discern unique signatures embedded in syntactic structures and lexical choices, which are influenced by specific training datasets. However, the proprietary nature of many LLMs restricts access to comparative data, posing significant hurdles. Additionally, the high degree of stylistic overlap among LLMs, especially those with similar architectures or trained on overlapping datasets, further complicates accurate classification. Continuous updates and fine-tuning of LLMs necessitate ongoing adjustments to attribution methodologies to account for evolving model characteristics [Wu et al., 2023b].

5. HUMAN-LLM CO-AUTHORED TEXT ATTRIBUTION

Besides creating text from scratch, LLMs are often used to extend sequences from human prompts. These perturbations have diminished the effectiveness of existing text detection methods [Bhat and Parthasarathy, 2020b]. Identifying text that combines input from both human authors and LLMs presents unique challenges. Hybrid texts may originate as human-written content, with LLMs employed for conditional generation, making it difficult to clearly distinguish between the stylistic features of human and machine contributions. There are fewer studies on this task due to its difficulty, and existing research often makes simplifications.

5.1 Problem Definition

A human-LLM co-authored text, also known as mixed text [Zhang et al., 2024] or collaborative human-AI writing [Richburg et al., 2024], is a piece of writing that is first created by a human and then revised or extended by LLMs, and vice versa. This task involves recognizing the nuances of multi-source authorship with fine-grained precision. Some studies categorize any text that is generated, modified, or extended by a machine as LLM-generated. This simplifies the task to either LLM-generated Text Detection or LLM-generated Text Attribution [Yang et al., 2023c; Crothers et al., 2023]. Other researchers handle LLM-revised human texts and human-revised LLM texts as a single category, alongside purely human-written and purely LLM-generated texts, approaching human-LLM co-authored text authorship attribution as a three-class classification problem [Zhang et al., 2024; Richburg et al., 2024]. One variation of this task is to detect the boundary between human-written and LLM-generated text [Cutler et al., 2021; Wang et al., 2024].

5.2 Methodologies

Human-authored texts tend to be more coherent and exhibit greater lexical diversity compared to LLM-generated texts [Guo et al., 2023; Dugan et al., 2023; Zhang et al., 2024]. Models like DNA-GPT [Yang et al., 2023a] and DetectGPT [Mitchell et al., 2023] utilized the T5 model [Raffel et al., 2020] to simulate scenarios in which humans modify LLM-generated texts. MIXSET [Zhang et al., 2024] offers a more comprehensive dataset that includes text refined by LLMs through polishing, completion, and rewriting operations.

To effectively analyze and classify human-LLM co-authored texts versus those solely authored by humans or LLMs, feature-based methods from the problem of LLM-generated text detection, such as Log-likelihood [Solaiman et al., 2019], GLTR [Gehrmann et al., 2019a], and log-rank [Mitchell et al., 2023], are adapted to this task. Additionally, neural network-based models such as BERT [Ippolito et al., 2020], Radar [Hu et al., 2023], and GPT-sentinel [Chen et al., 2023a] can also be applied. The complexity of this task increases as users may employ multiple LLMs to compose different sections of an article, further blurring the lines between human and machine-generated content. Consequently, the techniques used in the earlier detection of LLMs need to evolve continuously. This ongoing evolution in detection strategies mirrors the increasing sophistication of LLM outputs and the collaborative nature of modern text creation.

5.3 Open Challenges

Authorship attribution involving human-LLM co-authored pieces presents varying degrees of complexity, requiring different analytical approaches to accurately identify and differentiate the contributions of each author. For texts authored entirely by humans or LLMs, stylometric techniques can be utilized effectively. Human-authored texts often feature unique stylistic nuances, such as variable sentence structures and emotive language [Zhang et al., 2024]. In contrast, LLM-generated texts typically exhibit consistent syntax and a broader vocabulary [Guo et al., 2023]. Feature-based methods used in LLM-generated text detection can be adapted to classify texts by identifying these distinct patterns, thus attributing texts to their correct source.

Analyzing and classifying texts co-authored by humans and LLMs presents a significant challenge due to the blending of human and machine stylistic features. These texts often start as human drafts and are later extended or revised by LLMs, or the process might occur in reverse. This integration of styles creates a hybrid form that makes it difficult to distinguish distinct authorial markers, thereby complicating the attribution process.

Human-LLM co-authored texts pose a more intricate challenge due to the blending of stylistic and linguistic elements from both human authors and LLMs. These texts may begin as human drafts that are later extended or revised by LLMs, or vice versa, resulting in an integration of styles that obscures authorial markers [Liu et al., 2023b]. Advanced techniques are required to dissect these integrations, identifying where and how the contributions of LLM intersect with human input [Wang et al., 2024]. This involves detecting subtle shifts in style and contextual cues that indicate the extent of LLM involvement, allowing for accurate segmentation and attribution of authorship within hybrid documents.

6. RESOURCES AND EVALUATION METRICS

This section provides an in-depth examination of widely used benchmarks, datasets, and evaluation metrics in authorship attribution research, along with guidelines for selecting appropriate ones. These resources range from purely human-written texts to those generated by LLMs and human-LLM co-authored texts. This diversity is crucial for training and evaluating models across various tasks.

Traditional datasets focus exclusively on texts written by humans, while modern datasets include LLM-generated text, addressing the need to detect and attribute texts produced by LLMs. Additionally, this section covers commercial and open-source detectors commonly used to identify machine-generated text. Lastly, we summarize the common evaluation metrics employed in this field.

6.1 Benchmarks and Datasets

Authorship datasets encompass a wide range of sources, from formal literature to informal online communications, highlighting the increasing importance of user-generated content on social media. Human authorship datasets should include author identifiers and ideally contain multiple texts for each author. Manually collecting data is time-consuming and costly, motivating researchers to utilize web data sources such as Wikipedia and Reddit. In contrast, custom datasets for LLM-generated text are easier and more affordable to create, and are often built alongside human-written text to maintain a similar domain and format.

A general guideline for selecting and constructing datasets involves incorporating variations in domain, model architecture, and decoding strategies. Addressing class imbalance is crucial, as LLM-generated and human-written texts are often disproportionate. For human authorship data, it is recommended to choose content created before the widespread use of LLMs (GPT-3 [Brown et al., 2020] was released in June 2020, and ChatGPT followed in November 2022) to ensure that the texts were predominantly human-written.

Factors influencing the performance of existing authorship attribution models include the size of the training text [Hirst and Feiguina, 2007; Marton et al., 2005], the number of candidate authors [Koppel et al., 2006], and the imbalanced distribution of training texts among candidate authors [Stamatatos, 2008]. The availability of digital text in formats such as tweets, blogs, and articles has increased exponentially, providing more training data to accelerate the development of authorship attribution. However, the rapid growth of online communication has also led to shifts in writing behavior, resulting in shorter, fragmented, and less coherent social media tweets and text messages. For example, tweets are limited to 280 characters, whereas legal judgment documents contain thousands of words [Seroussi et al., 2011]. The challenge in social media stems from the brief nature of posts and a large pool of potential authors, making the attribution of short documents particularly difficult [Aborisade and Anwar, 2018; Seroussi et al., 2014; Theophilo et al., 2021].

Table 1 provides a comprehensive overview of 21 widely used benchmarks and datasets. These datasets are characterized by various statistics, including domain, size, word length, language, and the LLMs used to generate text. All listed datasets support LLM-generated text detection (Problem 2). However, fewer support LLM-generated text attribution (Problem 3) and Human-LLM Co-authored Text Attribution (Problem 4). These benchmarks often originate from human-written datasets such as XSum [Narayan et al., 2018], OpenWebText [Gokaslan and Cohen, 2019], and Wikipedia. Since Problems 2, 3, and 4 frequently treat human-written text as a single category for simplicity, rather than identifying individual authors as in Problem 1, many of these datasets are unsuitable for Problem 1, which requires unique author identification. Therefore, the representative datasets for human authorship (Problem 1) are summarized as follows:

- Amazon Review [Ni et al., 2019]: Featuring reviews with ratings, text, helpfulness votes, product metadata, and related links, this dataset provides a comprehensive view of consumer opinions, ideal for commercial authorship attribution studies.

Name	Domain	Size	Length	Language	Model	P2	P3	P4
TuringBench [Uchendu et al., 2021]	News	168,612 (5.2%)	100 to 400 words	en	GPT-1,2,3, GROVER, CTRL, XLM, XLNET, FAIR, TRANSFORMER-XL, PPLM	✓	✓	
TweepFake [Fagni et al., 2021]	Social media	25,572 (50.0%)	less than 280 characters	en	GPT-2, RNN, Markov, LSTM, CharRNN	✓		
ArguGPT [Liu et al., 2023c]	Academic essays	8,153 (49.5%)	300 words on average	en	GPT2-Xl, text-babbage-001, text-curie-001, davinci-001,002,003, GPT-3.5-Turbo	✓		
AuTexTification [Sarvazyan et al., 2023]	Tweets, reviews, news, legal, and how-to articles	163,306 (42.5%)	20 to 100 tokens	en, es	BLOOM, GPT-3	✓	✓	
CHEAT [Yu et al., 2023]	Academic paper abstracts	50,699 (30.4%)	163.9 words on average	en	ChatGPT	✓		
GPABench2 [Liu et al., 2023b]	Academic paper abstracts	2.385M (6.3%)	70 to 350 words	en	ChatGPT	✓		✓
Ghostbuster [Verma et al., 2023]	News, student essays, creative writing	23,091 (87.0%)	77 to 559 (median words per document)	en	ChatGPT, Claude	✓		
HC3 [Guo et al., 2023]	Reddit, Wikipedia, medicine, finance	125,230 (64.5%)	25 to 254 words	en, zh	ChatGPT	✓		
HC3 Plus [Su et al., 2023a]	News, social media	214,498	N/A	en, zh	ChatGPT	✓		
HC-Var [Xu et al., 2023]	News, reviews, essays, QA	144k (68.8%)	50 to 200 words	en	ChatGPT	✓		
HANSEN [Tripto et al., 2023]	Transcripts of speech (spoken text), statements (written text)	535k (96.1%)	less than 1k tokens	en	ChatGPT, PaLM2, Vicuna-13B	✓	✓	
M4 [Wang et al., 2023]	Wikipedia, WikiHow, Reddit, QA, news, paper abstracts, peer reviews	147,895 (24.2%)	more than 1k characters	ar, bg, en, id, ru, ur, zh	davinci-003, ChatGPT, GPT-4, Cohere, Dolly2, BLOOMz	✓		
MGTBench [He et al., 2023]	News, student essays, creative writing	21k (14.3%)	1 to 500 words	en	ChatGPT, ChatGLM, Dolly, GPT4All, StableLM, Claude	✓	✓	
MULTITuDE [Macko et al., 2023]	News	74,081 (10.8%)	200 to 512 tokens	ar, ca, cs, de, en, es, nl, pt, ru, uk, zh	GPT-3,4, ChatGPT, Llama-65B, Alpaca-LoRa-30B, Vicuna-13B, OPT-66B, OPT-IML-Max-1.3B	✓		
OpenGPTText [Chen et al., 2023a]	OpenWebText	58,790 (50.0%)	less than 2k words	en	ChatGPT	✓		
OpenLLMText [Chen et al., 2023b]	OpenWebText	344,530 (20%)	512 tokens	en	ChatGPT, PaLM, Llama, GPT2-XL	✓	✓	
Scientic Paper [Mosca et al., 2023]	Scientific papers	29k (55.2%)	900 tokens on average	en	SCIgen, GPT-2,3, ChatGPT, Galactica	✓		
RAID [Dugan et al., 2024]	News, Wikipedia, paper abstracts, recipes, Reddit, poems, book summaries, movie reviews	523,985 (2.9%)	323 tokens on average	cs, de, en	GPT-2,3,4, ChatGPT, Mistral-7B, MPT-30B, Llama2-70B, Cohere command and chat	✓		
M4GT-Bench [Wang et al., 2024]	Wikipedia, Wikihow, Reddit, arXiv abstracts, academic paper reviews, student essays	5,368,998 (96.6%)	more than 50 characters	ar, bg, de, en, id, it, ru, ur, zh	ChatGPT, davinci-003, GPT-4, Cohere, Dolly-v2, BLOOMz	✓	✓	✓
MAGE [Li et al., 2023a]	Reddit, reviews, news, QA, story writing, Wikipedia, academic paper abstracts	448,459 (34.4%)	263 words on average	en	GPT, Llama, GLM-130B, FLAN-T5 OPT, T0, BLOOM-7B1, GPT-J-6B, GPT-NeoX-2	✓		
MIXSET [Zhang et al., 2024]	Email, news, game reviews, academic paper abstracts, speeches, blogs	3.6k (16.7%)	50 to 250 words	en	GPT-4, Llama2	✓		✓

Table 1: Summary of Authorship Attribution Datasets and Benchmarks with LLM-Generated Text. Size is shown as the sum of LLM-generated and human-written texts (with the percentage of human-written texts in parentheses). Language is displayed using the two-letter ISO 639 abbreviation. Columns P2, P3, and P4 indicate whether the dataset supports problems described in Problem 2, 3, and 4, respectively.

Detector	Price	API	Website
GPTZero	150k words at $10/month, 10k words for free per month	Yes	https://gptzero.me/
ZeroGPT	100k characters for $9.99, 15k characters for free	Yes	https://www.zerogpt.com/
Sapling	50k characters for $25, 2k characters for free	Yes	https://sapling.ai/ai-content-detector
Originality.AI	200k words at $14.95/month	Yes	https://originality.ai/
CopyLeaks	300k words at $7.99/month	Yes	https://copyleaks.com/ai-content-detector
Winston	80k words at $12/month	Yes	https://gowinston.ai/
GPT Radar	$0.02/100 tokens	N/A	https://gptradar.com/
Turnitin's AI detector	License required	N/A	https://www.turnitin.com/solutions/topics/ai-writing/ai-detector/
GPT-2 Output Detector	Free	N/A	https://github.com/openai/gpt-2-output-dataset/tree/master/detector
Crossplag	Free	N/A	https://crossplag.com/ai-content-detector/
CatchGPT	Free	N/A	https://www.catchgpt.ai/
Quil.org	Free	N/A	https://aiwritingcheck.org/
Scribbr	Free	N/A	https://www.scribbr.com/ai-detector/
Draft Goal	Free	N/A	https://detector.dng.ai/
Writefull	Free	Yes	https://x.writefull.com/gpt-detector
Phrasly	Free	Yes	https://phrasly.ai/ai-detector
Writer	Free	Yes	https://writer.com/ai-content-detector/

Table 2: Overview of LLM-Generated Text Detectors.

- Aston 100 Idiolects Corpus [Heini and Kredens, 2021]: Comprising emails, essays, text messages, and business memos from 100 individuals (ages 18–22, native English speakers), this corpus provides a broad spectrum of text types for analyzing both content and stylistic features.

- Blog Authorship Corpus [Schler et al., 2006]: Contains over 680,000 blog posts from more than 19,000 authors, with an average of 35 posts per author. The texts, averaging 79 tokens, are informal and conversational.

- Deceptive Opinion Spam [Ott et al., 2011]: Includes 400 genuine and 400 deceptive hotel reviews, with deceptive reviews generated using Amazon Mechanical Turk, useful for studying the nuances of fake versus real reviews.

- Enron Email [Klimt and Yang, 2004]: Includes around 500,000 messages from 160 employees, offering long texts and high text-per-author variance, making it ideal for studying corporate communication styles.

- Fanfiction: Collected from fanfiction.net, this dataset includes fan-written fiction [Bischoff et al., 2020; Kestemont et al., 2021], providing insights into creative writing and authorship attribution in fictional narratives.

- IMDb1M [Seroussi et al., 2014]: Features over 270,000 movie reviews by 22,000 authors, with an average of 12.3 texts per author and an average text length of 121 tokens, suitable for analyzing shorter, user-generated content.

- Pushshift Reddit [Baumgartner et al., 2020]: This dataset comprises posts and comments from various subreddits, covering diverse topics and writing styles, making it suitable for analyzing informal online discourse.

- PAN [Kestemont et al., 2021; Bevendorff et al., 2022]: Offered by PAN workshops for benchmarking authorship attribution models and are used in various authorship attribution competitions.

- VALLA [Tyo et al., 2022]: Designed for benchmarking authorship attribution models, VALLA standardizes a range of texts across various genres and writing styles.

Table 2 summarizes various commercial and open-source LLM-generated text detectors. These detectors are primarily designed for LLM-generated text detection (Problem 2). Some detectors like GPTZero [Tian and Cui, 2023] can also detect human-LLM co-authored text and identify portions likely to be LLM-generated, addressing the boundary detection sub-problem of Problem 4 [Cutler et al., 2021]. Although many of these detectors claim over 99% accuracy in detecting LLM-generated text, few are tested on shared benchmark datasets. Despite their high accuracy claims, many detectors suffer from high false positive rates, which could falsely accuse individuals of plagiarism and undermine credibility and trust in genuine authors. Additionally, these detectors often lack robustness against variations in sampling strategies, adversarial attacks, and unseen domains and language models [Dugan et al., 2024].

6.2 Evaluation Metrics

Evaluation metrics such as the F1 score and AUCROC are essential for quantifying the performance of authorship models, providing a standardized means to assess and compare the effectiveness of different authorship attribution approaches. As in other classification tasks, existing studies predominantly use the Area Under the Receiver Operating Characteristic (AUCROC) and F1 score to evaluate attribution algorithms. In human authorship attribution, where there are a large number of candidate authors, retrieval metrics such as Mean Reciprocal Rank (MRR) and recall-at-k are used [Rivera-Soto et al., 2021]. Additionally, Self-BLEU are useful metrics, with a lower score indicating higher textual diversity [Zhang et al., 2024]. Common evaluation metrics include:

- Accuracy: Measures the proportion of correctly identified authors. High accuracy indicates that the model is effective at correctly classifying authors.

- Precision, Recall, and F1-Score: Crucial in imbalanced datasets. Precision indicates the relevance of identified instances, recall measures the ability to identify all relevant instances, and the F1-Score balances both.

- Area Under the Receiver Operating Characteristic Curve (AUCROC): Represents the trade-off between true positive rates and false positive rates, where higher values indicate better performance.

- False Positive Rate (FPR) and False Negative Rate (FNR): Critical for minimizing misclassification, with FPR measuring incorrect classification of human texts as LLM-generated and FNR the reverse.

- Recall-at-k: measure the probability that the correct author appears among the top k results when ranking targets by cosine similarity to a query text.

- Mean Absolute Error (MAE): Used to evaluate the performance of human-machine text boundary detection. It measures the average absolute difference between the predicted position index and the actual change point.

7. OPPORTUNITIES AND FUTURE DIRECTIONS

This section explores future directions in the field of authorship attribution, focusing on leveraging the potential of LLMs while addressing associated challenges. Future efforts should aim for finer granularity in authorship attribution, leveraging LLM capabilities, improving generalization, enhancing explainability, preventing misuse, developing standardized benchmarks, and integrating interdisciplinary perspectives to enrich the field.

7.1 Finer Granularity

Current authorship attribution methods face limitations when handling a more extensive range of candidate human authors or LLMs, presenting opportunities for future research. For instance, existing approaches for LLM-generated Text Attribution typically manage only a limited number of authors or models, which restricts their applicability in real-world scenarios where the pool of potential authors or models can be vast. Previous studies have often oversimplified the problem by categorizing all human-written text into a single category [Uchendu et al., 2021; He et al., 2023]. This approach ignores the diversity among human authors and fails to leverage the rich set of characteristics that distinguish individual writing styles. Future work can build upon traditional research on human authorship to develop methods capable of attributing human-written text to individual authors even within the context of LLM-generated content. This refinement will improve the accuracy and utility of authorship attribution models, especially in mixed datasets containing both human-written and LLM-generated texts.

Similarly, for Human-LLM Co-authored Text Attribution, there is a need to attribute text more precisely to individual human authors or specific LLMs. Current work simplifies human-written text, LLM-generated content, and texts co-authored by humans and LLMs into three broad categories, without differentiating between individual human authors and specific LLMs [Zhang et al., 2024; Richburg et al., 2024]. This approach overlooks the nuanced contributions of each author or model. By improving the granularity of attribution, future models can better distinguish between various human authors and LLMs, thus increasing the practicality and reliability of authorship attribution tools. Such advancements would be particularly valuable in collaborative environments where multiple human authors and LLMs contribute to a single body of work, enabling clearer recognition of each contributor's role.

7.2 Generalization

This subsection examines the applicability of current methodologies across varying LLMs, domains, genres, and languages. Domains refer to broad areas of knowledge or topics, while genres refer to specific styles or forms of writing within any domain. Domain generalization poses significant challenges due to variability in vocabularies, syntax, and styles across different subjects, complicating accurate authorship attribution. Attribution performance tends to drop when known and query texts differ in topic or genre [Altakrori et al., 2021]. Models like BERT and RoBERTa have shown limitations in cross-domain tasks [Barlas and Stamatatos, 2020b; Huertas-Tato et al., 2022], and adapting models to new domains remains difficult due to factors like dataset size variability and writing styles. Traditional methods focused on identifying less topic-dependent features, such as function words and part-of-speech n-grams [Madigan et al., 2005b; Menon and Choi, 2011], while recent approaches highlight the importance of training more powerful transformer-based models [Rivera-Soto et al., 2021] and techniques such as adversarial training [Ganin et al., 2016; Li et al., 2017; Ben-David et al., 2010; Ganin et al., 2016].

Genre generalization involves adapting to different writing styles, such as fiction, non-fiction, and poetry, each with unique features. Authors' adaptation to various genres dilutes their identifiable stylistic traits, complicating attribution. Similarly, distinguishing between human-written and LLM-generated text in different genres requires identifying genre-specific inconsistencies. The diversity of genres demands flexible models capable of understanding various narrative structures, tones, and stylistic elements. Adapting models to handle genre variations requires more advanced and flexible approaches for effective generalization. Current authorship attribution models also struggle with out-of-distribution issues when faced with languages and LLMs not encountered during training, leading to decreased accuracy and reliability [Koppel et al., 2005; Wu et al., 2023b]. Addressing this generalization problem is crucial for developing robust models that can handle diverse and evolving linguistic and model landscapes.

Improving generalization can potentially be achieved through several strategies. First, leveraging transfer learning by pre-training on large, diverse datasets and fine-tuning on specific domains enhances adaptability and performance [Barlas and Stamatatos, 2020a; Rodriguez et al., 2022b]. Second, developing domain- and genre-invariant features would improve robustness by focusing on core stylistic elements [Argamon et al., 2003]. Third, employing hybrid models and ensemble methods that integrate domain-specific knowledge can optimize prediction accuracy by drawing on the strengths of individual models [Bacciu et al., 2019]. Additionally, incorporating contextual factors such as the writing environment or intended audience, alongside data augmentation techniques, can bolster generalization. Finally, improving attribution in multilingual contexts enables models to operate effectively across various languages [Chen et al., 2022a; Shamardina et al., 2022]. As detectors could be biased against non-native English writers [Liang et al., 2023], enhancing multilingual generalization is crucial for fairness. Collectively, these approaches foster robust and adaptable models equipped to handle diverse styles and contexts.

7.3 Explainability

Improving explainability is crucial for ensuring transparency and trust in authorship attribution models as they become more integrated into fields such as law, academia, and journalism. Developing explainable techniques for authorship attribution can lead to more transparent methodologies, where the reasoning behind attributions is clear and understandable. Explainable authorship attribution can serve as evidence in legal proceedings [Chaski, 2005; Rocha et al., 2016]. Traditional attribution methods, which rely on stylistic and linguistic features to identify an author, struggle to distinguish between human-authored texts and those generated by LLMs, which adeptly replicate these features. This challenge requires improved methodologies that not only differentiate origins but also explore how LLMs emulate specific authorial styles [Boenninghoff et al., 2019a; Danilevsky et al., 2020].

Despite attempts such as analyzing internal attention weights or employing interpretation tools and visualization techniques [Wallace et al., 2019], word-level explanations are insufficient. The challenge remains to provide higher-level explainability that aligns with human cognitive processes [Rudin, 2019]. Advances in this area may include leveraging discourse-level relations and training models with human explanations for common sense reasoning to improve the explanatory depth of model-generated attributions [Rajani et al., 2019]. For instance, Kowalczyk et al. [2022] detected GPT-2-generated fake reviews using Shapley Additive Explanations (SHAP) [Lundberg and Lee, 2017].

7.4 Misuse Prevention

Future research should focus on refining existing authorship attribution methods to detect and prevent malicious activities such as generating misinformation, plagiarism, and propaganda [Goldstein et al., 2023; Hazell, 2023; Spitale et al., 2023; Lund et al., 2023]. These methods analyze stylistic features to detect discrepancies in claimed authorship and trace the origins of content, thereby identifying suspicious texts. For plagiarism, attribution models can compare writing styles with a database of known authors. In combating misinformation and propaganda, these models could identify and flag content patterns typical of known propagandists.

To ensure effectiveness in real-world tasks, authorship attribution models should be robust against out-of-domain data and adversarial attacks. Adversarial attacks including alternative spellings, article deletion, paragraph additions, case changes, zero-width spaces, whitespace manipulation, homoglyphs, number swaps, misspellings, paraphrasing, and synonym substitution, have been shown to effectively degrade detector performance [Dugan et al., 2024]. Diversifying training data with various writing styles and topics could improve robustness. Adversarial robustness could be achieved through adversarial training and employing ensemble methods to build resilience against intentional manipulations.

7.5 Leveraging LLM Capabilities

Leveraging LLMs can enhance both traditional feature-based stylometry methods and LLM-based approaches. By integrating LLMs with existing methods, researchers can gain deeper insight into stylistic nuances, improving the robustness of authorship detection across various textual genres and lengths. The increasingly large context length of LLMs enables in-context learning (ICL) [Brown et al., 2020] by incorporating more documents, enhancing the model's ability to capture intricate writing patterns.

Another promising approach is Retrieval-Augmented Generation (RAG) [Lewis et al., 2020]. RAG can enhance authorship attribution by retrieving additional documents for each author, thereby assisting in generating more contextually accurate results. Moreover, leveraging LLMs for data augmentation and synthetic data generation can create diverse training datasets, which in turn improves the generalization of attribution models [Albalak et al., 2024].

Combining text detection with other modalities, such as images, videos, or metadata, can potentially improve the accuracy and reliability of authorship attribution. Cross-modal analysis enables the integration of various data types, providing a more comprehensive view of the content and its origins. This holistic approach not only enhances the attribution process but also paves the way for innovative methodologies that are more resilient to the evolving nature of digital content.

7.6 Developing Standardized Benchmarks

The diversity in datasets and evaluation metrics currently hinders the comparability and generalizability of different authorship attribution methods. Establishing comprehensive benchmarks that encompass a wide range of text types and sources, including human-authored, LLM-generated, and human-LLM coauthored texts, would significantly enhance the field. Unified benchmarks should incorporate diverse text corpora from various genres, lengths, and languages to reflect the breadth of real-world applications. Clear evaluation metrics are essential for providing consistent and transparent measures of attribution accuracy, robustness, and computational efficiency, enabling fair comparisons between different models. Benchmarks should also include datasets that blend human-written and machine-generated content to simulate realistic tasks and test the robustness of attribution models.

To ensure ongoing relevance and challenge for attribution methods, benchmarks must be regularly updated to include new types of LLMs and detectors. By developing and adopting standardized benchmarks, the research community can foster more rigorous, reproducible, and comparable studies. This will ultimately drive advancements in authorship attribution methodologies and applications. These standardized benchmarks would serve as a foundation for the systematic evaluation of attribution techniques, promoting innovation and progress in addressing the complexities of authorship attribution in a rapidly evolving digital landscape.

7.7 Integrating Interdisciplinary Perspectives

Authorship attribution is inherently multidisciplinary, encompassing elements of linguistics, computer science, forensic science, and psychology [Stamatatos, 2009]. Future research should continue fostering collaboration across these fields to integrate diverse perspectives. This integrative approach can lead to innovative solutions and a deeper understanding of the challenges and potential of authorship attribution. Combining insights from various disciplines can foster the creation of holistic attribution models that account for both the intricacies of human language and the complexities of LLM-generated texts. Such collaboration could also spearhead initiatives to standardize evaluation metrics for authorship attribution tools, ensuring their effectiveness across diverse contexts and compliance with ethical standards.

Linguistics can dissect textual structures and stylistic nuances, identifying unique linguistic fingerprints of authors. It explores novel features that enhance robustness across domains and improve explainability. Forensic science contributes through technological tools and methodologies, enabling a precise examination of physical and digital texts. Psychology, particularly psycholinguistics, provides insight into how the brain processes function words and grammatical markers distinctively from lexical content words, revealing correlations with socio-cultural categories such as gender, age, and native language [Chambers et al., 2013; Nerbonne, 2014; Seals and Shalin, 2023], which are pivotal in understanding identity and social affiliations [Argamon et al., 2009]. The Linguistic Inquiry and Word Count (LIWC) tool exemplifies how automated text analysis can use more than 100 psychological dimensions to analyze word use, reflecting distinct language variations among different groups in specific genres and languages [Goldstein-Stewart et al., 2009; Pennebaker et al., 2015; Dudău and Sava, 2021].

Combining these interdisciplinary perspectives enhances our ability to distinguish between human- and machine-generated texts, addressing the emerging challenges posed by sophisticated language models. By integrating linguistic theory with advanced computational techniques, forensic methodologies, and psychological insights, researchers can develop more comprehensive and nuanced authorship attribution frameworks. These frameworks will be better equipped to handle the diverse range of writing styles and contexts, ultimately leading to more accurate and reliable attribution outcomes. Furthermore, interdisciplinary collaboration can drive the development of ethical guidelines and best practices, ensuring that authorship attribution is conducted responsibly and with respect for individuals' privacy and rights.

8. ETHICAL AND PRIVACY CONCERNS

In the evolving landscape of authorship attribution, it is crucial to prioritize ethical considerations to safeguard privacy, integrity, and the rightful ownership of content. The attribution of text to specific authors or models raises significant ethical and privacy issues. Misattribution can lead to wrongful accusations or misinterpretation of an author's intent [Lund et al., 2023]. Additionally, the use of attribution technologies must balance the need for accountability with respect to individuals' privacy and the potential for misuse in surveilling or censoring content.

Authorship attribution techniques are essential in digital forensics, cybersecurity, and plagiarism detection. However, the potential to reveal the identities of anonymous authors presents significant ethical challenges. Applications such as linking user accounts across platforms and identifying compromised accounts raise privacy concerns and ethical questions about surveillance and profiling individuals based on their writing style.

The use of authorship attribution methods must be carefully managed to protect individual privacy and adhere to ethical standards, particularly in sensitive areas such as journalism, political dissent, and corporate whistle-blowing [Sison et al., 2023]. Ensuring that these methods are not used to undermine privacy rights or expose individuals to risks without their consent is essential. Despite existing measures to prohibit the unethical use of LLMs, these restrictions could be evaded through prompt engineering and jail-breaking, posing risks of phishing and fraud scams.

Furthermore, the increasing difficulty in distinguishing between human and LLM-generated content raises concerns about intellectual property, plagiarism, and accountability. Accurate attribution is crucial for maintaining academic and creative integrity, yet tools and methods for achieving this must evolve rapidly to keep up with technological advancements. The deployment of LLMs in generating content across various domains—from journalism to literature—necessitates a rethinking of authorship norms and the legal frameworks governing creative works.

9. CONCLUSION

The field of authorship attribution is experiencing both unprecedented challenges and remarkable opportunities with the advent of LLMs. Whether the objective is to identify human authors, differentiate between human- and machine-generated texts, attribute texts to specific LLMs, or manage the complexities of human-LLM co-authored texts, ongoing innovation is imperative. Effectively addressing these multifaceted issues requires interdisciplinary approaches and collaborative efforts among researchers. This survey explores various problems within authorship attribution, offering a comprehensive comparison of methodologies and datasets. By integrating robustness, explainability, and interdisciplinary perspectives, we highlight the importance of developing methods that are not only accurate but also socially relevant and trustworthy. We highlight the strengths and limitations of current approaches, identify key open problems, and outline future research directions. This holistic analysis equips researchers and practitioners with the knowledge necessary to navigate the evolving landscape of authorship attribution, emphasizing critical areas for future research and development.

Acknowledgments

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 17STQAC00001-07-04, NSF awards (SaTC-2241068, IIS-2339198, and POSE-2346158), a Cisco Research Award, and a Microsoft Accelerate Foundation Models Research Award. The views and conclusions contained in this document are

those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security and the National Science Foundation.

References

Sahar Abdelnabi and Mario Fritz. 2021. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 121–140.

Opeyemi Aborisade and Mohd Anwar. 2018. Classification for authorship of tweets by comparing logistic regression and naive bayes classifiers. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. IEEE, 269–276.

Sadia Afroz, Michael Brennan, and Rachel Greenstadt. 2012. Detecting hoaxes, frauds, and deception in writing style online. In *2012 IEEE Symposium on Security and Privacy*. IEEE, 461–475.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. 2024. A survey on data selection for language models. *ArXiv preprint* abs/2402.16827 (2024). https://arxiv.org/abs/2402.16827

Malik Altakrori, Jackie Chi Kit Cheung, and Benjamin C. M. Fung. 2021. The Topic Confusion Task: A Novel Evaluation Scenario for Authorship Attribution. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 4242–4256. https://doi.org/10.18653/v1/2021.findings-emnlp.359

Nicholas Andrews and Marcus Bishop. 2019. Learning Invariant Representations of Social Media Users. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 1684–1695. https://doi.org/10.18653/v1/D19-1178

Shlomo Argamon. 2018. Computational forensic authorship analysis: Promises and pitfalls. *Language and Law/Linguagem e Direito* 5, 2 (2018), 7–37.

Shlomo Argamon, Moshe Koppel, Jonathan Fine, and Anat Rachel Shimoni. 2003. Gender, genre, and writing style in formal written texts. *Text & talk* 23, 3 (2003), 321–346.

Shlomo Argamon, Moshe Koppel, James W Pennebaker, and Jonathan Schler. 2009. Automatically profiling the author of an anonymous text. *Commun. ACM* 52, 2 (2009), 119–123.

Mikhail J Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. 2001. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding: 4th International Workshop, IH 2001 Pittsburgh, PA, USA, April 25–27, 2001 Proceedings 4*. Springer, 185–200.

Andrea Bacciu, Massimo La Morgia, Alessandro Mei, Eugenio Nerio Nemmi, Valerio Neri, and Julinda Stefa. 2019. Cross-Domain Authorship Attribution Combining Instance Based and Profile-Based Features.. In *CLEF (Working Notes)*.

Douglas Bagnall. 2015. Author identification using multi-headed recurrent neural networks. *ArXiv preprint* abs/1506.04891 (2015). https://arxiv.org/abs/1506.04891

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *ArXiv preprint* abs/2310.05130 (2023). https://arxiv.org/abs/2310.05130

Sylvio Barbon, Rodrigo Augusto Igawa, and Bruno Bogaz Zarpelão. 2017. Authorship verification applied to detection of compromised accounts on online social networks: A continuous approach. *Multimedia Tools and Applications* 76 (2017), 3213–3233.

Georgios Barlas and Efstathios Stamatatos. 2020a. Cross-domain authorship attribution using pre-trained language models. In *Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part I 16*. Springer, 255–266.

Georgios Barlas and Efstathios Stamatatos. 2020b. Cross-domain authorship attribution using pre-trained language models. In *Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part I 16*. Springer, 255–266.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, Vol. 14. 830–839.

Jonas Becker, Jan Philip Wahle, Terry Ruas, and Bela Gipp. 2023. Paraphrase detection: Human vs. machine content. *ArXiv preprint* abs/2303.13989 (2023). https://arxiv.org/abs/2303.13989

Nicole Mariah Sharon Belvisi, Naveed Muhammad, and Fernando Alonso-Fernandez. 2020. Forensic authorship analysis of microblogging texts using n-grams and stylometric features. In *2020 8th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 1–6.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79 (2010), 151–175.

Daria Beresneva. 2016. Computer-generated text detection using machine learning: A systematic review. In *Natural Language Processing and Information Systems: 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016, Salford, UK, June 22-24, 2016, Proceedings 21*. Springer, 421–426.

Janek Bevendorff, Berta Chulvi, Elisabetta Fersini, Annina Heini, Mike Kestemont, Krzysztof Kredens, Maximilian Mayerl, Reynier Ortega-Bueno, Piotr Pęzik, Martin Potthast, et al. 2022. Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, and Style Change Detection. In *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, 382–394.

Meghana Moorthy Bhat and Srinivasan Parthasarathy. 2020a. How Effectively Can Machines Defend Against Machine-Generated Fake News? An Empirical Study. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*. Association for Computational Linguistics, Online, 48–53. https://doi.org/10.18653/v1/2020.insights-1.7

Meghana Moorthy Bhat and Srinivasan Parthasarathy. 2020b. How Effectively Can Machines Defend Against Machine-Generated Fake News? An Empirical Study. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*. Association for Computational Linguistics, Online, 48–53. https://doi.org/10.18653/v1/2020.insights-1.7

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. Conda: Contrastive domain adaptation for ai-generated text detection. *ArXiv preprint* abs/2309.03992 (2023). https://arxiv.org/abs/2309.03992

Sebastian Bischoff, Niklas Deckers, Marcel Schliebs, Ben Thies, Matthias Hagen, Efstathios Stamatatos, Benno Stein, and Martin Potthast. 2020. The importance of suppressing domain style in authorship analysis. *ArXiv preprint* abs/2005.14714 (2020). https://arxiv.org/abs/2005.14714

Benedikt T. Boenninghoff, Steffen Hessler, Dorothea Kolossa, and Robert M. Nickel. 2019a. Explainable Authorship Verification in Social Media via Attention-based Similarity Learning. In *2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, December 9-12, 2019*. IEEE, 36–45. https://doi.org/10.1109/BigData47090.2019.9005650

Benedikt T. Boenninghoff, Robert M. Nickel, Steffen Zeiler, and Dorothea Kolossa. 2019b. Similarity Learning for Authorship Verification in Social Media. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*. IEEE, 2457–2461. https://doi.org/10.1109/ICASSP.2019.8683405

Ilker Nadi Bozkurt, Ozgur Baghoglu, and Erkan Uyar. 2007. Authorship attribution. In *2007 22nd international symposium on computer and information sciences*. IEEE, 1–5.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

Florian Cafiero and Jean-Baptiste Camps. 2023. Who could be behind QAnon? Authorship attribution with supervised machine-learning. *ArXiv preprint* abs/2303.02078 (2023). https://arxiv.org/abs/2303.02078

Shuyang Cai and Wanyun Cui. 2023. Evade ChatGPT detectors via a single space. *ArXiv preprint* abs/2307.02599 (2023). https://arxiv.org/abs/2307.02599

Megha Chakraborty, SM Tonmoy, SM Zaman, Krish Sharma, Niyar R Barman, Chandan Gupta, Shreya Gautam, Tanay Kumar, Vinija Jain, Aman Chadha, et al. 2023b. Counter Turing Test CTˆ 2: AI-Generated Text Detection is Not as Easy as You May Think–Introducing AI Detectability Index. *ArXiv preprint* abs/2310.05030 (2023). https://arxiv.org/abs/2310.05030

Souradip Chakraborty, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2023a. On the possibilities of ai-generated text detection. *ArXiv preprint* abs/2304.04736 (2023). https://arxiv.org/abs/2304.04736

Jack K Chambers, Peter Trudgill, and Natalie Schilling-Estes. 2013. *The handbook of language variation and change*. Wiley Online Library.

Carole E Chaski. 2005. Who's at the keyboard? Authorship attribution in digital evidence investigations. *International journal of digital evidence* 4, 1 (2005), 1–13.

Canyu Chen and Kai Shu. 2024a. Can LLM-Generated Misinformation Be Detected?. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=ccxD4mtkTU

Canyu Chen and Kai Shu. 2024b. Combating misinformation in the age of LLMs: Opportunities and challenges. *AI Magazine* (2024). https://doi.org/10.1002/aaai.12188

Canyu Chen, Haoran Wang, Matthew Shapiro, Yunyu Xiao, Fei Wang, and Kai Shu. 2022b. Combating health misinformation in social media: Characterization, detection, intervention, and open issues. *ArXiv preprint* abs/2211.05289 (2022). https://arxiv.org/abs/2211.05289

Xingyuan Chen, Peng Jin, Siyuan Jing, and Chunming Xie. 2022a. Automatic detection of Chinese generated essayss based on pre-trained BERT. In *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Vol. 10. IEEE, 2257–2260.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023a. Gpt-sentinel: Distinguishing human and chatgpt generated content. *ArXiv preprint* abs/2305.07969 (2023). https://arxiv.org/abs/2305.07969

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023b. Token prediction as implicit classification to identify LLM-generated text. *ArXiv preprint* abs/2311.08723 (2023). https://arxiv.org/abs/2311.08723

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All That's 'Human' Is Not Gold: Evaluating Human Evaluation of Generated Text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 7282–7296. https://doi.org/10.18653/v1/2021.acl-long.565

Evan Crothers, Nathalie Japkowicz, and Herna L Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access* (2023).

Joseph Cutler, Liam Dugan, Shreya Havaldar, and Adam Stein. 2021. Automatic Detection of Hybrid Human-Machine Text Boundaries. (2021).

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A Survey of the State of Explainable AI for Natural Language Processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, Suzhou, China, 447–459. https://aclanthology.org/2020.aacl-main.46

Edwin Dauber, Rebekah Overdorf, and Rachel Greenstadt. 2017. Stylometric authorship attribution of collaborative documents. In *Cyber Security Cryptography and Machine Learning: First International Conference, CSCML 2017, Beer-Sheva, Israel, June 29-30, 2017, Proceedings 1*. Springer, 115–135.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Diana Paula Dudău and Florin Alin Sava. 2021. Performing multilingual analysis with Linguistic Inquiry and Word Count 2015 (LIWC2015). An equivalence study of four languages. *Frontiers in Psychology* 12 (2021), 570568.

Liam Dugan, Alyssa Hwang, Filip Trhlik, Josh Magnus Ludan, Andrew Zhu, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A Shared Benchmark for Robust Evaluation of Machine-Generated Text Detectors. *ArXiv preprint* abs/2405.07940 (2024). https://arxiv.org/abs/2405.07940

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 12763–12771.

Maciej Eder. 2015. Does size matter? Authorship attribution, small samples, big problem. *Digital Scholarship in the Humanities* 30, 2 (2015), 167–182.

Maël Fabien, Esau Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. 2020. BertAA : BERT fine-tuning for Authorship Attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*. NLP Association of India (NLPAI), Indian Institute of Technology Patna, Patna, India, 127–137. https://aclanthology.org/2020.icon-main.16

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. TweepFake: About detecting deepfake tweets. *Plos one* 16, 5 (2021), e0251415.

Vitalii Fishchuk and Daniel Braun. 2023. Efficient Black-Box Adversarial Attacks on Neural Text Detectors. *ArXiv preprint* abs/2311.01873 (2023). https://arxiv.org/abs/2311.01873

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling GPT-2, GPT-3 and Grover. *PeerJ Computer Science* 7 (2021), e443.

Rinaldo Gagiano, Maria Myung-Hee Kim, Xiuzhen Zhang, and Jennifer Biggs. 2021. Robustness Analysis of Grover for Machine-Generated News Detection. In *Proceedings of the The 19th Annual Workshop of the Australasian Language Technology Association*. Australasian Language Technology Association, Online, 119–127. https://aclanthology.org/2021.alta-1.12

Matthias Gallé, Jos Rozen, Germán Kruszewski, and Hady Elsahar. 2021. Unsupervised and distributional detection of machine-generated text. *ArXiv preprint* abs/2111.02878 (2021). https://arxiv.org/abs/2111.02878

Margherita Gambini, Tiziano Fagni, Fabrizio Falchi, and Maurizio Tesconi. 2022. On pushing DeepFake Tweet Detection capabilities to the limits. In *14th ACM Web Science Conference 2022*. 154–163.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research* 17, 1 (2016), 2096–2030.

Catherine A Gao, Frederick M Howard, Nikolay S Markov, Emma C Dyer, Siddhi Ramesh, Yuan Luo, and Alexander T Pearson. 2022. Comparing scientific abstracts generated by ChatGPT to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers. *BioRxiv* (2022), 2022–12.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019a. GLTR: Statistical Detection and Visualization of Generated Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Florence, Italy, 111–116. https://doi.org/10.18653/v1/P19-3019

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019b. GLTR: Statistical Detection and Visualization of Generated Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Florence, Italy, 111–116. https://doi.org/10.18653/v1/P19-3019

Aaron Gokaslan and Vanya Cohen. 2019. OpenWebText Corpus. http://Skylion007.github.io/OpenWebTextCorpus

Josh A. Goldstein, Girish Sastry, Micah Musser, Renee DiResta, Matthew Gentzel, and Katerina Sedova. 2023. Generative Language Models and Automated Influence Operations: Emerging Threats and Potential Mitigations. arXiv:2301.04246 [cs.CY]

Jade Goldstein-Stewart, Ransom Winder, and Roberta Sabin. 2009. Person Identification from Text and Speech Genre Samples. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. Association for Computational Linguistics, Athens, Greece, 336–344. https://aclanthology.org/E09-1039

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6572

Tim Grant. 2007. Quantifying evidence in forensic authorship analysis. *International Journal of Speech, Language & the Law* 14, 1 (2007).

Tim Grant. 2020. Text messaging forensics: Txt 4n6: idiolect-free authorship analysis? In *The Routledge handbook of forensic linguistics*. Routledge, 558–575.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *ArXiv preprint* abs/2301.07597 (2023). https://arxiv.org/abs/2301.07597

Oren Halvani and Lukas Graner. 2021. Posnoise: An effective countermeasure against topic biases in authorship analysis. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*. 1–12.

Hans WA Hanley and Zakir Durumeric. 2024. Machine-made media: Monitoring the mobilization of machine-generated articles on misinformation and mainstream news websites. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 18. 542–556.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting LLMs With Binoculars: Zero-Shot Detection of Machine-Generated Text. *ArXiv preprint* abs/2401.12070 (2024). https://arxiv.org/abs/2401.12070

Julian Hazell. 2023. Large language models can be used to effectively scale spear phishing campaigns. *ArXiv preprint* abs/2305.06972 (2023). https://arxiv.org/abs/2305.06972

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. Mgtbench: Benchmarking machine-generated text detection. *ArXiv preprint* abs/2303.14822 (2023). https://arxiv.org/abs/2303.14822

Pezik-P. Heini, A. and K. Kredens. 2021. GPTZero: Towards detection of AI-generated text using zero-shot and supervised methods. http://fold.aston.ac.uk/handle/123456789/17

Graeme Hirst and Ol'ga Feiguina. 2007. Bigrams of syntactic labels for authorship discrimination of short texts. *Literary and Linguistic Computing* 22, 4 (2007), 405–417.

David I Holmes. 1994. Authorship attribution. *Computers and the Humanities* 28 (1994), 87–106.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=rygGQyrFvH

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2023. Semstamp: A semantic watermark with paraphrastic robustness for text generation. *ArXiv preprint* abs/2310.03991 (2023). https://arxiv.org/abs/2310.03991

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 328–339. https://doi.org/10.18653/v1/P18-1031

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in Neural Information Processing Systems* 36 (2023), 15077–15095.

Baixiang Huang, Canyu Chen, and Kai Shu. 2024. Can Large Language Models Identify Authorship? *ArXiv preprint* abs/2403.08213 (2024). https://arxiv.org/abs/2403.08213

Javier Huertas-Tato, Alvaro Huertas-Garcia, Alejandro Martin, and David Camacho. 2022. PART: Pre-trained Authorship Representation Transformer. *ArXiv preprint* abs/2209.15373 (2022). https://arxiv.org/abs/2209.15373

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic Detection of Generated Text is Easiest when Humans are Fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 1808–1822. https://doi.org/10.18653/v1/2020.acl-main.164

Zunera Jalil and Anwar M Mirza. 2009. A review of digital watermarking techniques for text documents. In *2009 International Conference on Information and Multimedia Technology*. IEEE, 230–234.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020a. Automatic Detection of Machine Generated Text: A Critical Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 2296–2309. https://doi.org/10.18653/v1/2020.coling-main.208

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020b. Automatic Detection of Machine Generated Text: A Critical Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 2296–2309. https://doi.org/10.18653/v1/2020.coling-main.208

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008*, Marc Najork, Andrei Z. Broder, and Soumen Chakrabarti (Eds.). ACM, 219–230. https://doi.org/10.1145/1341531.1341560

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, Valencia, Spain, 427–431. https://aclanthology.org/E17-2068

Mike Kestemont, Enrique Manjavacas, Ilia Markov, Janek Bevendorff, Matti Wiegmann, Efstathios Stamatatos, Martin Potthast, and Benno Stein. 2021. Overview of the Cross-Domain Authorship Verification Task at PAN 2021.. In *CLEF (Working Notes)*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning*. PMLR, 17061–17084.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *Machine Learning: ECML 2004: 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004. Proceedings 15*. Springer, 217–226.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.

Moshe Koppel, Jonathan Schler, Shlomo Argamon, and Eran Messeri. 2006. Authorship attribution with thousands of candidate authors. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 659–660.

Moshe Koppel, Jonathan Schler, and Eran Messeri. 2008. Authorship attribution in law enforcement scenarios. *NATO Security Through Science Series D-Information and Communication Security* 15 (2008), 111.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Determining an author's native language by mining a text for errors. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 624–628.

Peter Kowalczyk, Marco Röder, Alexander Dürr, and Frédéric Thiesse. 2022. Detecting and understanding textual deepfakes in online reviews. (2022).

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2024a. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems* 36 (2024).

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2024b. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems* 36 (2024).

Pranav Kulkarni, Ziqing Ji, Yan Xu, Marko Neskovic, and Kevin Nolan. 2023. Exploring Semantic Perturbations on Grover. *ArXiv preprint* abs/2302.00509 (2023). https://arxiv.org/abs/2302.00509

Tharindu Kumarage and Huan Liu. 2023. Neural Authorship Attribution: Stylometric Analysis on Large Language Models. *ArXiv preprint* abs/2308.07305 (2023). https://arxiv.org/abs/2308.07305

Tharindu Kumarage, Paras Sheth, Raha Moraffah, Joshua Garland, and Huan Liu. 2023. How reliable are ai-generated-text detectors? an assessment framework using evasive soft prompts. *ArXiv preprint* abs/2310.05095 (2023). https://arxiv.org/abs/2310.05095

Ksenia Lagutina, Nadezhda Lagutina, Elena Boychuk, Inna Vorontsova, Elena Shliakhtina, Olga Belyaeva, Ilya Paramonov, and PG Demidov. 2019. A survey on stylometric text features. In *2019 25th Conference of Open Innovations Association (FRUCT)*. IEEE, 184–195.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting Fake Content with Relative Entropy Scoring. *Pan* 8, 27-31 (2008), 4.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

Linyang Li, Pengyu Wang, Ke Ren, Tianxiang Sun, and Xipeng Qiu. 2023b. Origin tracing and detecting of llms. *ArXiv preprint* abs/2304.14072 (2023). https://arxiv.org/abs/2304.14072

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2023a. MAGE: Machine-generated Text Detection in the Wild. https://arxiv.org/abs/2305.13242

Zheng Li, Yu Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. End-to-End Adversarial Memory Network for Cross-domain Sentiment Classification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 2237–2243. https://doi.org/10.24963/ijcai.2017/311

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. GPT detectors are biased against non-native English writers. *Patterns* 4, 7 (2023).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint* abs/1907.11692 (2019). https://arxiv.org/abs/1907.11692

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023c. ArguGPT: evaluating, understanding and identifying argumentative essays generated by GPT models. *ArXiv preprint* abs/2304.07666 (2023). https://arxiv.org/abs/2304.07666

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2023a. On the Detectability of ChatGPT Content: Benchmarking, Methodology, and Evaluation through the Lens of Academic Writing. https://arxiv.org/abs/2306.05524

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2023b. On the detectability of chatgpt content: benchmarking, methodology, and evaluation through the lens of academic writing. *arXiv e-prints* (2023), arXiv–2306.

Ning Lu, Shengcai Liu, Rui He, Qi Wang, Yew-Soon Ong, and Ke Tang. 2023. Large language models can be guided to evade ai-generated text detection. *ArXiv preprint* abs/2305.10847 (2023). https://arxiv.org/abs/2305.10847

Brady D Lund, Ting Wang, Nishith Reddy Mannuru, Bing Nie, Somipam Shimray, and Ziang Wang. 2023. ChatGPT and a new academic reality: Artificial Intelligence-written research papers and the ethics of the large language models in scholarly publishing. *Journal of the Association for Information Science and Technology* 74, 5 (2023), 570–581.

Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 4765–4774. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Samuel Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, et al. 2023. MULTITuDE: Large-Scale Multilingual Machine-Generated Text Detection Benchmark. *ArXiv preprint* abs/2310.13606 (2023). https://arxiv.org/abs/2310.13606

Dominik Macko, Robert Moro, Adaku Uchendu, Ivan Srba, Jason Samuel Lucas, Michiharu Yamashita, Nafis Irtiza Tripto, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2024. Authorship obfuscation in multilingual machine-generated text detection. *ArXiv preprint* abs/2401.07867 (2024). https://arxiv.org/abs/2401.07867

David Madigan, Alexander Genkin, David D Lewis, Shlomo Argamon, Dmitriy Fradkin, and Li Ye. 2005b. Author identification on the large scale. In *Proceedings of the 2005 Meeting of the Classification Society of North America (CSNA)*.

David Madigan, Alexander Genkin, David D Lewis, and Dmitriy Fradkin. 2005a. Bayesian multinomial logistic regression for author identification. In *AIP conference proceedings*, Vol. 803. American Institute of Physics, 509–516.

Andrei Manolache, Florin Brad, Elena Burceanu, Antonio Barbalau, Radu Ionescu, and Marius Popescu. 2021. Transferring bert-like transformers' knowledge for authorship verification. *ArXiv preprint* abs/2112.05125 (2021). https://arxiv.org/abs/2112.05125

Yuval Marton, Ning Wu, and Lisa Hellerstein. 2005. On compression-based text classification. In *Advances in Information Retrieval: 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005. Proceedings 27*. Springer, 300–314.

Rohith Menon and Yejin Choi. 2011. Domain Independent Authorship Attribution without Domain Adaptation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*. Association for Computational Linguistics, Hissar, Bulgaria, 309–315. https://aclanthology.org/R11-1043

Hasan Mesut Meral, Bülent Sankur, A Sumru Özsoy, Tunga Güngör, and Emre Sevinç. 2009. Natural language watermarking via morphosyntactic alterations. *Computer Speech & Language* 23, 1 (2009), 107–125.

Sven Meyer zu Eissen, Benno Stein, and Marion Kulig. 2007. Plagiarism detection without reference collections. In *Advances in Data Analysis: Proceedings of the 30 th Annual Conference of the Gesellschaft für Klassifikation eV, Freie Universität Berlin, March 8–10, 2006*. Springer, 359–366.

Niloofar Mireshghallah, Justus Mattern, Sicun Gao, Reza Shokri, and Taylor Berg-Kirkpatrick. 2023. Smaller language models are better black-box machine-generated text detectors. *ArXiv preprint* abs/2305.09859 (2023). https://arxiv.org/abs/2305.09859

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *ArXiv preprint* abs/2301.11305 (2023). https://arxiv.org/abs/2301.11305

Edoardo Mosca, Mohamed Hesham Ibrahim Abdalla, Paolo Basso, Margherita Musumeci, and Georg Groh. 2023. Distinguishing Fact from Fiction: A Benchmark Dataset for Identifying Machine-Generated Scientific Papers in the LLM Era.. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*. 190–207.

Frederick Mosteller and David L Wallace. 1963. Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed Federalist Papers. *J. Amer. Statist. Assoc.* 58, 302 (1963), 275–309.

Shaoor Munir, Brishna Batool, Zubair Shafiq, Padmini Srinivasan, and Fareed Zaffar. 2021. Through the Looking Glass: Learning to Attribute Synthetic Text Generated by Language Models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Online, 1811–1822. https://doi.org/10.18653/v1/2021.eacl-main.155

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 1797–1807. https://doi.org/10.18653/v1/D18-1206

Tempestt Neal, Kalaivani Sundararajan, Aneez Fatima, Yiming Yan, Yingfei Xiang, and Damon Woodard. 2017. Surveying stylometry techniques and applications. *ACM Computing Surveys (CSuR)* 50, 6 (2017), 1–36.

John Nerbonne. 2014. The secret life of pronouns. what our words say about us. *Literary and Linguistic Computing* 29, 1 (2014), 139–142.

Hoang-Quoc Nguyen-Son, Ngoc-Dung T Tieu, Huy H Nguyen, Junichi Yamagishi, and Isao Echi Zen. 2017. Identifying computer-generated text using statistical analysis. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 1504–1511.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 188–197. https://doi.org/10.18653/v1/D19-1018

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 309–319. https://aclanthology.org/P11-1032

Ajay Patel, Delip Rao, and Chris Callison-Burch. 2023. Learning Interpretable Style Embeddings via Prompting LLMs. *ArXiv preprint* abs/2305.12696 (2023). https://arxiv.org/abs/2305.12696

James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. *The development and psychometric properties of LIWC2015*. Technical Report.

Steven T Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review* 21 (2014), 1112–1130.

Nektaria Potha and Efstathios Stamatatos. 2019. Improving author verification based on topic modeling. *Journal of the Association for Information Science and Technology* 70, 10 (2019), 1074–1088.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2023a. Deepfake text detection: Limitations and opportunities. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1613–1630.

Xiao Pu, Jingyu Zhang, Xiaochuang Han, Yulia Tsvetkov, and Tianxing He. 2023b. On the zero-shot generalization of machine-generated text detectors. *ArXiv preprint* abs/2310.05165 (2023). https://arxiv.org/abs/2310.05165

Chen Qian, Tianchang He, and Rao Zhang. 2017. Deep learning based authorship identification. *Report, Stanford University* (2017), 1–9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67. http://jmlr.org/papers/v21/20-074.html

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, 4932–4942. https://doi.org/10.18653/v1/P19-1487

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. 2022. Effect of scale on catastrophic forgetting in neural networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=GhVS8_yPeEa

Aquia Richburg, Calvin Bao, and Marine Carpuat. 2024. Automatic Authorship Analysis in Human-AI Collaborative Writing. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 1845–1855.

Rafael A. Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y. Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. 2021. Learning Universal Authorship Representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 913–919. https://doi.org/10.18653/v1/2021.emnlp-main.70

Anderson Rocha, Walter J Scheirer, Christopher W Forstall, Thiago Cavalcante, Antonio Theophilo, Bingyu Shen, Ariadne RB Carvalho, and Efstathios Stamatatos. 2016. Authorship attribution for social media forensics. *IEEE transactions on information forensics and security* 12, 1 (2016), 5–33.

Juan Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022a. Cross-Domain Detection of GPT-2-Generated Technical Text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 1213–1233. https://doi.org/10.18653/v1/2022.naacl-main.88

Juan Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022b. Cross-Domain Detection of GPT-2-Generated Technical Text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 1213–1233. https://doi.org/10.18653/v1/2022.naacl-main.88

Sebastian Ruder, Parsa Ghaffari, and John G Breslin. 2016. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *ArXiv preprint* abs/1609.06686 (2016). https://arxiv.org/abs/1609.06686

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence* 1, 5 (2019), 206–215.

Joseph Rudman. 1997. The state of authorship attribution studies: Some problems and solutions. *Computers and the Humanities* 31 (1997), 351–365.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *ArXiv preprint* abs/2303.11156 (2023). https://arxiv.org/abs/2303.11156

Chakaveh Saedi and Mark Dras. 2021. Siamese networks for large-scale author identification. *Computer Speech & Language* 70 (2021), 101241.

Joni Salminen, Chandrashekhar Kandpal, Ahmed Mohamed Kamel, Soon-gyo Jung, and Bernard J Jansen. 2022. Creating and detecting fake reviews of online products. *Journal of Retailing and Consumer Services* 64 (2022), 102771.

Areg Mikael Sarvazyan, José Ángel González, Marc Franco-Salvador, Francisco Rangel, Berta Chulvi, and Paolo Rosso. 2023. Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains. *ArXiv preprint* abs/2309.11285 (2023). https://arxiv.org/abs/2309.11285

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging.. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, Vol. 6. 199–205.

Ipek Baris Schlicht and Angel Felipe Magnossão de Paula. 2021. Unified and multilingual author profiling for detecting haters. *ArXiv preprint* abs/2109.09233 (2021). https://arxiv.org/abs/2109.09233

Tal Schuster, Roei Schuster, Darsh J. Shah, and Regina Barzilay. 2020. The Limitations of Stylometry for Detecting Machine-Generated Fake News. *Computational Linguistics* 46, 2 (2020), 499–510. https://doi.org/10.1162/coli_a_00380

SM Seals and Valerie L Shalin. 2023. Long-form analogies generated by chatGPT lack human-like psycholinguistic properties. *ArXiv preprint* abs/2306.04537 (2023). https://arxiv.org/abs/2306.04537

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do Massively Pretrained Language Models Make Better Storytellers?. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Association for Computational Linguistics, Hong Kong, China, 843–861. https://doi.org/10.18653/v1/K19-1079

Yanir Seroussi, Russell Smyth, and Ingrid Zukerman. 2011. Ghosts from the high court's past: Evidence from computational linguistics for Dixon ghosting for Mctiernan and rich. *University of New South Wales Law Journal, The* 34, 3 (2011), 984–1005.

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2014. Authorship Attribution with Topic Models. *Computational Linguistics* 40, 2 (2014), 269–310. https://doi.org/10.1162/COLI_a_00173

Danish Shakeel and Nitin Jain. 2021. Fake news detection and fact verification using knowledge graphs and machine learning. *ResearchGate preprint* 10 (2021).

Tatiana Shamardina, Vladislav Mikhailov, Daniil Chernianskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022. Findings of the the ruatd shared task 2022 on artificial text detection in russian. *ArXiv preprint* abs/2206.01583 (2022). https://arxiv.org/abs/2206.01583

Abhay Sharma, Ananya Nandan, and Reetika Ralhan. 2018. An investigation of supervised learning methods for authorship attribution in short hinglish texts using char & word n-grams. *ArXiv preprint* abs/1812.10281 (2018). https://arxiv.org/abs/1812.10281

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. 2024. Red teaming language model detectors with language models. *Transactions of the Association for Computational Linguistics* 12 (2024), 174–189.

Prasha Shrestha, Sebastian Sierra, Fabio González, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017a. Convolutional Neural Networks for Authorship Attribution of Short Texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, Valencia, Spain, 669–674. https://aclanthology.org/E17-2106

Prasha Shrestha, Sebastian Sierra, Fabio González, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017b. Convolutional Neural Networks for Authorship Attribution of Short Texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, Valencia, Spain, 669–674. https://aclanthology.org/E17-2106

Kai Shu, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. Mining disinformation and fake news: Concepts, methods, and recent advancements. *Disinformation, misinformation, and fake news in social media: Emerging research challenges and opportunities* (2020), 1–19.

Kai Shu, Suhang Wang, Jiliang Tang, Reza Zafarani, and Huan Liu. 2017. User identity linkage across online social networks: A review. *Acm Sigkdd Explorations Newsletter* 18, 2 (2017), 5–17.

Richard Sinnott and Zijian Wang. 2021. Linking user accounts across social media platforms. In *2021 IEEE/ACM 8th International Conference on Big Data Computing, Applications and Technologies (BDCAT'21)*. 18–27.

Alejo Jose G Sison, Marco Tulio Daza, Roberto Gozalo-Brizuela, and Eduardo C Garrido-Merchán. 2023. ChatGPT: more than a "weapon of mass deception" ethical challenges and responses from the human-centered artificial intelligence (HCAI) perspective. *International Journal of Human–Computer Interaction* (2023), 1–20.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *ArXiv preprint* abs/1908.09203 (2019). https://arxiv.org/abs/1908.09203

Irene Solaiman, Zeerak Talat, William Agnew, Lama Ahmad, Dylan Baker, Su Lin Blodgett, Canyu Chen, Hal Daumé III, Jesse Dodge, Isabella Duan, et al. 2023. Evaluating the social impact of generative ai systems in systems and society. *arXiv preprint arXiv:2306.05949* (2023).

Rafael Rivera Soto, Kailin Koch, Aleem Khan, Barry Chen, Marcus Bishop, and Nicholas Andrews. 2024. Few-Shot Detection of Machine-Generated Text using Style Representations. *ArXiv preprint* abs/2401.06712 (2024). https://arxiv.org/abs/2401.06712

Giovanni Spitale, Nikola Biller-Andorno, and Federico Germani. 2023. AI model GPT-3 (dis) informs us better than humans. *Science Advances* 9, 26 (2023), eadh1850.

Efstathios Stamatatos. 2008. Author identification: Using text sampling to handle the class imbalance problem. *Information Processing & Management* 44, 2 (2008), 790–799.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology* 60, 3 (2009), 538–556.

Efstathios Stamatatos. 2016. Authorship verification: a review of recent advances. *Research in Computing Science* 123 (2016), 9–25.

Efstathios Stamatatos and Moshe Koppel. 2011. Plagiarism and authorship analysis: introduction to the special issue. *Language Resources and Evaluation* 45 (2011), 1–4.

Harald Stiff and Fredrik Johansson. 2022. Detecting computer-generated disinformation. *International Journal of Data Science and Analytics* 13, 4 (2022), 363–383.

Ariel Stolerman, Rebekah Overdorf, Sadia Afroz, and Rachel Greenstadt. 2014. Breaking the closed-world assumption in stylometric authorship attribution. In *Advances in Digital Forensics X: 10th IFIP WG 11.9 International Conference, Vienna, Austria, January 8-10, 2014, Revised Selected Papers 10*. Springer, 185–205.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023b. DetectLLM: Leveraging Log Rank Information for Zero-Shot Detection of Machine-Generated Text. *ArXiv preprint* abs/2306.05540 (2023). https://arxiv.org/abs/2306.05540

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023c. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *ArXiv preprint* abs/2306.05540 (2023). https://arxiv.org/abs/2306.05540

Zhenpeng Su, Xing Wu, Wei Zhou, Guangyuan Ma, and Songlin Hu. 2023a. Hc3 plus: A semantic-invariant human chatgpt comparison corpus. *ArXiv preprint* abs/2309.02731 (2023). https://arxiv.org/abs/2309.02731

Kalaivani Sundararajan and Damon Woodard. 2018. What represents "style" in authorship attribution?. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2814–2822. https://aclanthology.org/C18-1238

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023a. The science of detecting llm-generated texts. *ArXiv preprint* abs/2303.07205 (2023). https://arxiv.org/abs/2303.07205

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023b. The science of detecting llm-generated texts. *ArXiv preprint* abs/2303.07205 (2023). https://arxiv.org/abs/2303.07205

Antonio Theophilo, Romain Giot, and Anderson Rocha. 2021. Authorship attribution of social media messages. *IEEE Transactions on Computational Social Systems* (2021).

Edward Tian and Alexander Cui. 2023. GPTZero: Towards detection of AI-generated text using zero-shot and supervised methods. https://gptzero.me

Mercan Topkara, Cuneyt M Taskiran, and Edward J Delp III. 2005. Natural language watermarking. In *Security, Steganography, and Watermarking of Multimedia Contents VII*, Vol. 5681. SPIE, 441–452.

Nafis Irtiza Tripto, Adaku Uchendu, Thai Le, Mattia Setzu, Fosca Giannotti, and Dongwon Lee. 2023. HANSEN: human and AI spoken text benchmark for authorship analysis. *ArXiv preprint* abs/2310.16746 (2023). https://arxiv.org/abs/2310.16746

Jacob Tyo, Bhuwan Dhingra, and Zachary C Lipton. 2022. On the state of the art in authorship attribution and authorship verification. *ArXiv preprint* abs/2209.06869 (2022). https://arxiv.org/abs/2209.06869

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship Attribution for Neural Text Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 8384–8395. https://doi.org/10.18653/v1/2020.emnlp-main.673

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A Benchmark Environment for Turing Test in the Age of Neural Text Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 2001–2016. https://doi.org/10.18653/v1/2021.findings-emnlp.172

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*. Association for Computational Linguistics, Tokyo, Japan, 355–368. https://doi.org/10.18653/v1/W19-8643

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2023. Gpt-who: An information density-based machine-generated text detector. *ArXiv preprint* abs/2310.06202 (2023). https://arxiv.org/abs/2310.06202

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2023. Ghostbuster: Detecting text ghostwritten by large language models. *ArXiv preprint* abs/2305.15047 (2023). https://arxiv.org/abs/2305.15047

Bertie Vidgen, Adarsh Agrawal, Ahmed M Ahmed, Victor Akinwande, Namir Al-Nuaimi, Najla Alfaraj, Elie Alhajjar, Lora Aroyo, Trupti Bavalatti, Borhane Blili-Hamelin, et al. 2024. Introducing v0. 5 of the ai safety benchmark from mlcommons. *arXiv preprint arXiv:2404.12241* (2024).

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP Interpret: A Framework for Explaining Predictions of NLP Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*. Association for Computational Linguistics, Hong Kong, China, 7–12. https://doi.org/10.18653/v1/D19-3002

Wenxiao Wang, Alexander Levine, and Soheil Feizi. 2022. Improved Certified Defenses against Data Poisoning with (Deterministic) Finite Aggregation. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 22769–22783. https://proceedings.mlr.press/v162/wang22m.html

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohanned Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, et al. 2024. M4GT-Bench: Evaluation Benchmark for Black-Box Machine-Generated Text Detection. *to appear in ACL 2024* (2024).

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, et al. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *ArXiv preprint* abs/2305.14902 (2023). https://arxiv.org/abs/2305.14902

Anna Wegmann, Marijn Schraagen, and Dong Nguyen. 2022. Same Author or Just Same Topic? Towards Content-Independent Style Representations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*. Association for Computational Linguistics, Dublin, Ireland, 249–268. https://doi.org/10.18653/v1/2022.repl4nlp-1.26

Jana Winter. 2019. Exclusive: FBI document warns conspiracy theories are a new domestic terrorism threat. *Yahoo News* 1 (2019).

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F Wong, and Lidia S Chao. 2023b. A survey on llm-gernerated text detection: Necessity, methods, and future directions. *ArXiv preprint* abs/2310.14724 (2023). https://arxiv.org/abs/2310.14724

Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. 2023a. Dipmark: A stealthy, efficient and resilient watermark for large language models. *ArXiv preprint* abs/2310.07710 (2023). https://arxiv.org/abs/2310.07710

Han Xu, Jie Ren, Pengfei He, Shenglai Zeng, Yingqian Cui, Amy Liu, Hui Liu, and Jiliang Tang. 2023. On the Generalization of Training-based ChatGPT Detection Methods. arXiv:2310.01307 [cs.CL]

Xianjun Yang, Wei Cheng, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023a. DNA-GPT: Divergent N-Gram Analysis for Training-Free Detection of GPT-Generated Text. *ArXiv preprint* abs/2305.17359 (2023). https://arxiv.org/abs/2305.17359

Xianjun Yang, Wei Cheng, Yue Wu, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023b. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *ArXiv preprint* abs/2305.17359 (2023). https://arxiv.org/abs/2305.17359

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Petzold, William Yang Wang, and Wei Cheng. 2023c. A survey on detection of llms-generated content. *ArXiv preprint* abs/2310.15654 (2023). https://arxiv.org/abs/2310.15654

Xi Yang, Jie Zhang, Kejiang Chen, Weiming Zhang, Zehua Ma, Feng Wang, and Nenghai Yu. 2022. Tracing Text Provenance via Context-Aware Lexical Substitution. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 11613–11621. https://ojs.aaai.org/index.php/AAAI/article/view/21415

Peipeng Yu, Jiahan Chen, Xuan Feng, and Zhihua Xia. 2023. Cheat: A large-scale dataset for detecting chatgpt-written abstracts. *ArXiv preprint* abs/2304.12008 (2023). https://arxiv.org/abs/2304.12008

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending Against Neural Fake News. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 9051–9062. https://proceedings.neurips.cc/paper/2019/hash/3e9f0fc9b2f89e043bc6233994dfcf76-Abstract.html

Haolan Zhan, Xuanli He, Qiongkai Xu, Yuxiang Wu, and Pontus Stenetorp. 2023. G3Detector: General GPT-generated text detector. *ArXiv preprint* abs/2305.12680 (2023). https://arxiv.org/abs/2305.12680

Qihui Zhang, Chujie Gao, Dongping Chen, Yue Huang, Yixin Huang, Zhenyang Sun, Shilin Zhang, Weiye Li, Zhengyan Fu, Yao Wan, et al. 2024. LLM-as-a-Coauthor: Can Mixed Human-Written and Machine-Generated Text Be Detected?. In *Findings of the Association for Computational Linguistics: NAACL 2024*. 409–436.

Richong Zhang, Zhiyuan Hu, Hongyu Guo, and Yongyi Mao. 2018. Syntax Encoding with Application in Authorship Attribution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 2742–2753. https://doi.org/10.18653/v1/D18-1294

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable robust watermarking for ai-generated text. *ArXiv preprint* abs/2306.17439 (2023). https://arxiv.org/abs/2306.17439

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Neural Deepfake Detection with Factual Structure of Text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 2461–2470. https://doi.org/10.18653/v1/2020.emnlp-main.193

George Kingsley Zipf. 2016. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio books.

Exploring Large Language Models for Feature Selection: A Data-centric Perspective

Dawei Li*
Arizona State University
Tempe, AZ, USA
daweili5@asu.edu

Zhen Tan*
Arizona State University
Tempe, AZ, USA
ztan36@asu.edu

Huan Liu
Arizona State University
Tempe, AZ, USA
huanliu@asu.edu

ABSTRACT

The rapid advancement of Large Language Models (LLMs) has significantly influenced various domains, leveraging their exceptional few-shot and zero-shot learning capabilities. In this work, we aim to explore and understand the LLMs-based feature selection methods from a data-centric perspective. We begin by categorizing existing feature selection methods with LLMs into two groups: data-driven feature selection which requires numerical values of samples to do statistical inference and text-based feature selection which utilizes prior knowledge of LLMs to do semantical associations using descriptive context. We conduct experiments in both classification and regression tasks with LLMs in various sizes (e.g., GPT-4, ChatGPT and LLaMA-2). Our findings emphasize the effectiveness and robustness of text-based feature selection methods and showcase their potentials using a real-world medical application. We also discuss the challenges and future opportunities in employing LLMs for feature selection, offering insights for further research and development in this emerging field.

1. INTRODUCTION

Recent years have witnessed the remarkable development of Large Language Models (LLMs) [1; 4; 53; 58] across various domains and areas [37; 6; 36; 3]. By leveraging extensive training corpora and well-designed prompting strategies, LLMs demonstrate impressive few-shot and zero-shot capabilities in diverse tasks such as question answering [65; 64; 57], information extraction [60] and knowledge discovery [46; 63; 62]. The tuning-free nature also makes in-context learning (ICL) in LLMs achieve a great balance between efficiency and effectiveness [54].

Feature selection [10; 35] is a critical data serving step that ensures relevant and high-quality data for downstream machine learning and data mining applications. While existing data-driven selection methods have achieved great success in scenarios with abundant data and metadata, there is an increasing demand for efficient feature selection methods with few or even zero samples for various reasons [72]. This need is particularly pronounced in sensitive applications such as predicting survival times for cancer patients [56; 66], where privacy concerns may prevent hospitals and patients from sharing their data, posing difficulties in the feature selection and engineering process. To address this challenge, recent studies [26; 23] have explored leveraging the few-shot capability in LLMs to perform feature selection in low-resource settings and got promising results.

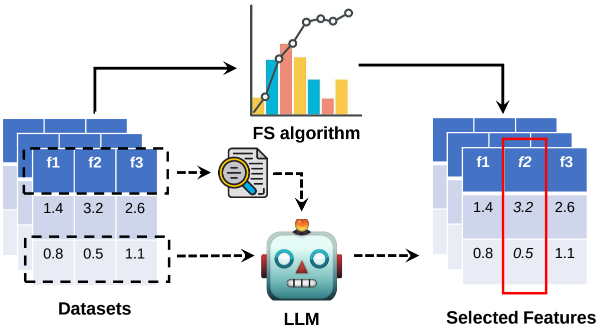


Figure 1: Comparison of traditional feature selection (FS) algorithms and LLM-based methods. Instead of requiring the whole dataset to make statistic inference, recent works prompt LLMs to select features in an efficient way. This is often achieved in a (*i*) data-driven, or (*ii*) text-based way.

In this work, our objective is to thoroughly explore and understand LLMs-based feature selection methods from a data-centric perspective. The conclusions and insights drawn from this exploration can provide insightful guidance for real-world applications where different types of resources and data are available. To begin with, we categorize the prompting strategies in previous studies [8; 26; 40; 23] for LLMs-based feature selection into two groups: (*i*) data-driven methods, which provide specific samples to LLMs [40; 23], and (*ii*) text-based methods, which incorporate feature and task descriptions into the instruction [8; 26]. These two prompting strategies require different data types: data-driven methods rely on sample points from datasets to do statistical inference while text-based methods need descriptive context for better semantic association between features and target variables. Figure 1 presents an overall comparison between the abovementioned methods and traditional feature selection algorithms. These differences make us curious about how LLMs perform with each of them under different data availability settings.

We conduct extensive experiments to explore the two methods in both classification and regression tasks with different LLMs in various sizes (E.g. GPT-4, ChatGPT and LLaMA-2). ***A key finding*** based on the results is that, text-based

*Equal Constributions

feature selection using LLMs is more effective and stable across various low-resource settings. Additionally, it shows a more pronounced scaling law with respect to the size of LLMs compared to data-driven approaches. Furthermore, we carried out a comparative evaluation between text-based feature selection using LLMs and traditional feature selection methods. ***A general observation*** is that, the text-based approach is relatively more robust and competitive across different resource availability settings.

Based on the abovementioned findings, we further explore the *applicability* of text-based feature selection with LLMs in a medical application. Specifically, we focus on the prediction of survival time for cancer patients [56; 66], which is a crucial task to evaluate both patient health and treatment effectiveness. To enhance the LLMs' understanding of medical-specific gene names, we developed a **R**etrieval-**A**ugmented **F**eature **S**election **(RAFS)** method that leverages descriptions from the National Institutes of Health (NIH) as auxiliary context. Experiment results demonstrate our RAFS's effectiveness in performing effective feature selection while safeguarding patient's privacy. Finally, we outline the existing challenges and potential opportunities in employing LLMs for feature selection.

To summarize, our contributions in this work are as follows:

- We propose a general taxonomy for the existing LLMs-based feature selection methods, splitting them into data-driven and text-based methods.
- Through an analysis under varying data availability conditions, we identify the strengths and weaknesses of these two methods, finding that text-based approaches are more effective and robust.
- We showcase the utilization of the text-based feature selection method with LLMs in a real-world medical application and introduce RAFS, a method designed to handle domain-specific feature selection with LLMs.
- We systematically analyze the existing challenges and potential future directions for using LLMs in feature selection, providing further insights and guidelines for future studies.

2. RELATED WORK

2.1 Feature Selection

Feature selection is the process of identifying and selecting the most relevant and important features or variables from a dataset to improve the performance and efficiency of a machine learning model [10; 20; 5; 35]. These feature selection methods can be generally categorized into three groups: filter, wrapper, and embedded approaches. Filter methods [31] first rank features by performing correlation analysis and then selecting the most important ones for the following learning step. Typical filter methods include mutual information [32; 11], Fisher score [24; 19] and maximum mean discrepancy [52]. By contrast, wrapper methods [30] use heuristic search strategies to identify a feature subset that optimally enhances the performance of certain prediction models (e.g., sequential selection [42] and recursive feature elimination [21]). For embedded approaches, it works together with specific machine learning models in the training phase by adding various regularization items in the loss function to encourage feature sparsity [55; 71].

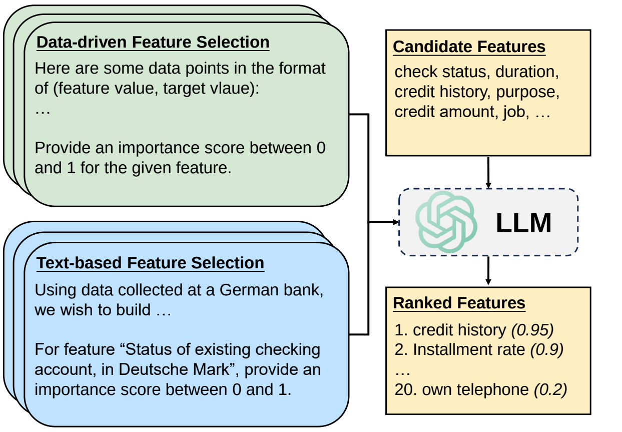

Data-driven Feature Selection
Here are some data points in the format of (feature value, target vlaue):
…
Provide an importance score between 0 and 1 for the given feature.
Text-based Feature Selection
Using data collected at a German bank, we wish to build …
For feature "Status of existing checking account, in Deutsche Mark", provide an importance score between 0 and 1.
Candidate Features
check status, duration, credit history, purpose, credit amount, job, …
LLM
Ranked Features
1. credit history (0.95)
2. Installment rate (0.9)
…
20. own telephone (0.2)

Figure 2: Prompting strategies for data-driven and text-based feature selection methods with LLMs.

2.2 Feature Selection with LLMs

There are already some works exploring the adaptation of LLMs in feature selection. [8] try to extract the relevant knowledge from LLMs as the task prior to performing feature selection, reinforcement learning and casual discovery. For feature selection, they design a prompt to instruct GPT-3 [4] to generate whether given features are important by answering "Yes" or "No". Following them, [26] expand the LLMs-based feature selection and propose three different pipelines that directly utilize the generated text output. They also conduct extensive experiments in evaluation across various model scales and prompting strategies. Besides, some studies devise more complex pipelines with LLMs in feature selection and feature engineering. [40] introduce an In-Context Evolutionary Search (ICE-SEARCH) in Medical Predictive Analytics (MPA) applications. It involves recurrently optimizing the selected features by prompting LLMs to perform feature filtering based on test scores. [23] employ LLMs as feature engineers to produce meta-features beyond the original features and combine them with simple machine learning models to improve predictions in downstream tasks. In this work, we aim to explore and understand LLMs in performing feature selection from a data perspective, offering further insights and hints for the adaptation of LLM-based feature selectors in real-world applications.

3. A DATA-CENTRIC TAXONOMY

Given a pre-trained LLM M, we follow the scoring-based method proposed by [26], which prompt M to generate an importance score s_i for the given feature/ concept f_i in the dataset d:

$$s_i = \mathrm{M}(P_{f_i}), \quad i \in \{1, ..., l\}, \tag{1}$$

where l is the total number of the features in dataset d. P_{f_i} refers to the specific prompt we use to generate the importance score. We will discuss two methods for constructing prompts in Sections 3.1 and 3.2, each focusing on different capabilities of LLMs. Figure 2 demonstrates the detailed prompting strategy for each of them.

3.1 Data-driven Feature Selection

Recently, LLMs have been employed to directly handle numeric data, demonstrating their capabilities in numerical prediction and analytics [18; 27]. Therefore, we build a data-driven feature selection method with LLMs by providing both features' value n_{f_i} and the value of the target variable n_y. Intuitively, LLMs are supposed to infer the correlation and perform statistical analysis to determine the importance of the given feature in the dataset.

To be more specific, assume there are m samples available in the dataset d, we first build the sample pairs SP_i using values of the i_{th} feature and target variable:

$$SP_i = \{(n^j_{f_i}, n^j_y)\}, \quad i \in \{1, ..., l\}, j \in \{1, ..., m\}. \tag{2}$$

Then, we curate the prompt P_{f_i} using SP_i as few-shot examples and other instruction context C:

$$P^{Data}_{f_i} = \text{prompt}(C, SP_i), \tag{3}$$

here *prompt* is a function to concatenate the information and build a fluent instruction for LLMs.

3.2 Text-based Feature Selection

Another line of work [8; 26] tries to employ the extensive semantics knowledge in LLMs [33] to perform feature selection. Specifically, they incorporate detailed dataset descriptions in the prompt, instructing LLMs to semantically distinguish the importance of a given feature using their inherent knowledge and experience.

In our studies, we consider two concrete descriptive contexts: dataset description (des_d) and feature description (des_{f_i}). The dataset description includes the task's objective, details about the dataset's collection, and an explanation of the target variable. The feature description focuses on detailing and clarifying the feature to be scored.

Formally, we build prompts by integrating the abovementioned information:

$$P^{Text}_{f_i} = \text{prompt}(C, des_d, des_{f_i}). \tag{4}$$

We give specific instruction examples for the two feature selection methods in Appendix A.

4. ANALYSES

4.1 Experiment Settings

In this section, we evaluate the performance of the LLM-based feature selection methods using various datasets and models.

Models. Below are the LLMs used in our experiment.

- LLaMA-2 [58]: 7B parameters.
- LLaMA-2 [58]: 13B parameters.
- ChatGPT [45]: ∼175B parameters[1].
- GPT-4 [1]: ∼1.7T parameters[1].

We use the “gpt-4-turbo-2024-04-09” and “gpt-3.5-turbo-0125 models via API calling. For LLaMA-2, we do local inference with the checkpoints available from Huggingface, namely “llama-2-70b-chat-hf” and “llama-2-13b-chat-hf”.

Compared Methods As the main methods to be analyzed in this section, we use “w/ data” and “w/ text” to represent the data-driven and text-based feature selection methods. We also compare the LLM-based feature selection methods with the following traditional feature selection baselines:

- Filtering by Mutual Information (MI) [32].
- Recursive Feature Elimination (RFE) [21].
- Minimum Redundancy Maximum Relevance selection (MRMR) [11].
- Random feature selection.

Dataset	# of samples	# of features
Adult	48842	14
Bank	45211	16
Communities	1994	102
Credit-g	1000	20
Heart	918	11
Myocardial	686	92
Diabetes	442	20
NBA	538	28
Rideshare	5000	18
Wine	6497	11

Table 1: Statistics of the datasets used.

Datasets. In our evaluation, we consider both classification and regression tasks. For the classification task, we use six datasets: Adult [2], Bank [44], Communities [50], Credit-g [28], Heart[2] and Myocardial [16]. For the regression task, we use four datasets: Diabetes [13], NBA[3], Rideshare[4] and Wine [2]. Detailed statistics of datasets are given in Table 1.

Implementation Details. For each dataset, we fix the feature selection ratio to 30%. We vary the data availability for evaluations with 16-shot, 32-shot, 64-shot, and 128-shot configurations. The test performance is measured using a downstream L2-penalized logistic/ linear regression model, selected via grid search with 5-fold cross-validation. We use the area under the ROC curve (AUROC) to evaluate classification tasks and mean absolute error (MAE) for regression.

4.2 Result Analysis

We present our main experimental results in Figure 3 and Figure 4 for analyzing, and highlighting the following findings for answering the RESEARCH QUESTION:

Finding 1: Text-based feature selection is more effective than data-driven ones with LLMs in low-resource settings. As results demonstrated in Figure 3 (a), almost in every LLM and task (except LLaMA-2-7B in classification), the performance of small machine learning models with the text-based feature selection method surpasses that of the data-driven feature selection method. This finding is consistent when we delve into feature selection methods' performance in each data availability, as depicted in Figure 4. Additionally, in Figure 3 (a), we notice for the same LLM, the text-based feature selection method

[1] ∼ denotes the estimated size [26] of closed-source LLMs

[2] https://kaggle.com/datasets/fedesoriano/heart-failure-prediction

[3] https://www.kaggle.com/datasets/bryanchungweather/nba-player-stats-dataset-for-the-2023-2024

[4] https://www.kaggle.com/datasets/aaronweymouth/nyc-rideshare-raw-data

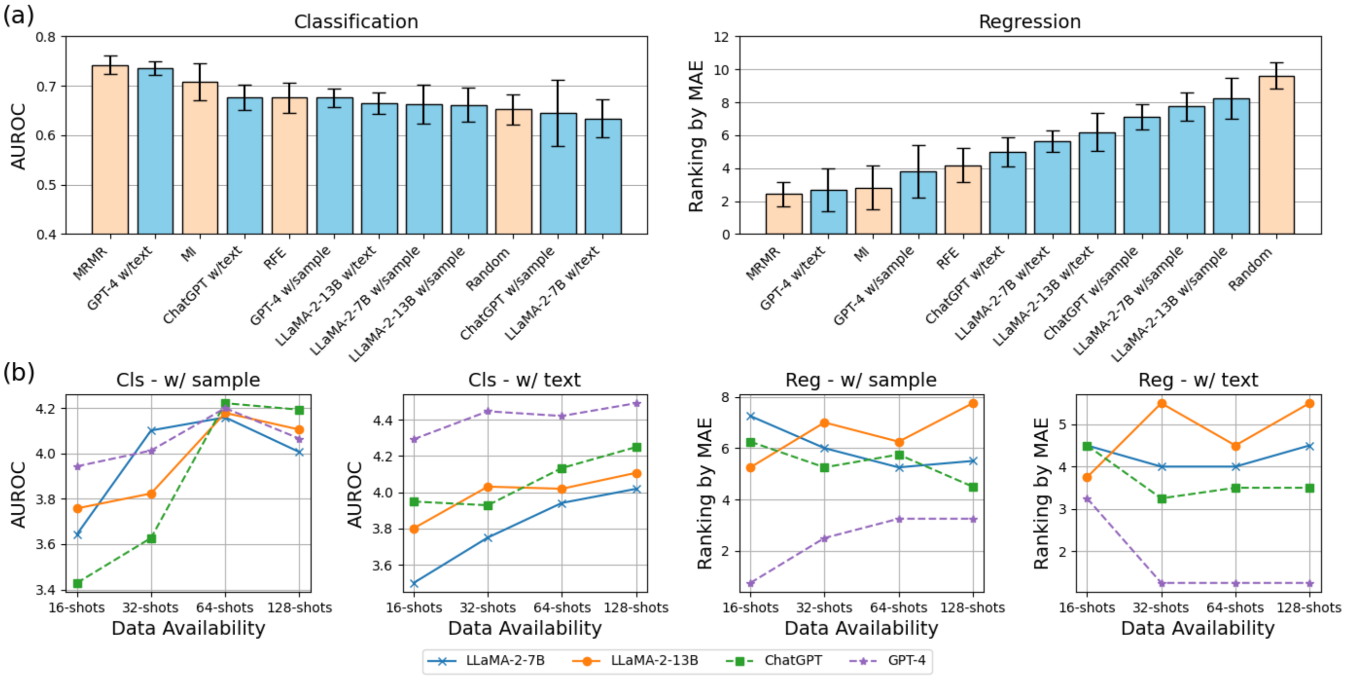


Figure 3: (a) Average AUROC (left; higher is better) and ranking by MAE (right; lower is better) across all datasets. (b) Each LLM's feature selection results, separated by task types (CLS and REG) and selection methods (w/sample and w/text).

usually leads to a smaller standard variant among various data availability settings. This further underscores the robustness and independence of the text-based feature selection method with respect to sample size.

	AUROC	Ranking by MAE
MI	0.779	**1.75**
RFE	0.758	3.50
MRMR	**0.798**	2.25
GPT-4 w/text	0.783	2.50

Table 2: Feature selection results in the full dataset with traditional data-driven methods and "GPT-4 w/text".

Finding 2: Text-based feature selection with the most advanced LLMs achieves comparable performance with traditional feature selection methods in every data availability setting. In Figure 3 (a), we observe that while GPT-4 with the text-based feature selection method performs slightly below the best traditional method (MRMR), it still demonstrates comparable performance, making it a competitive feature selection method in few-shot scenarios. However, when the LLM backbone is switched to less capable models, such as LLaMA, the text-based selection method shows a significant performance drop. Additionally, we experiment on the full dataset using 'GPT-4 w/text' alongside three traditional feature selection methods, and found that GPT-4 with the text-based method remains competitive even in the full-shot scenario.

Finding 3: Data-driven feature selection using LLMs struggles when number of samples increases. An interesting phenomenon we observed is a significant performance drop in the classification task when the sample size increases from 64 to 128 using the data-driven feature selection method (Figure 3 (b)). This drop is consistently observed across all four LLMs, indicating that each model generates poorer feature subsets as the sample size grows. We attribute this issue to LLMs struggling with processing long sequences, a challenge highlighted in many previous studies [12; 39]. This limitation constrains the effectiveness of data-driven feature selection, which is why we did not include it in the full-shot experiment.

Finding 4: Text-based feature selection exhibits a stronger scaling law with model size compared to data-driven feature selection with LLMs. We investigated how scaling laws in model size affect feature selection capabilities. In Figure 3 (b), we observe a clear correlation between the size of LLMs and their text-based feature selection capabilities. In contrast, while GPT-4 shows significantly superior performance in data-driven feature selection, the other three LLMs do not clearly follow the scaling law. This suggests that text-based feature selection is a reliable approach that can be enhanced by using powerful LLMs.

5. SURVIVAL PREDICTION - A CASE STUDY

We use a biomedical task to showcase the utilization of LLMs-based feature selection in real-world applications. Survival time prediction [56; 66] aims to predict cancer patients' survival time based on their physical and physiological indicators, playing a critical role in patient risk management and boosting treatment selection. One of the significant challenges in survival prediction datasets is the huge volume of features (e.g., there are around 20,000 gene expression features in the TCGA [56] dataset). While previous studies performed data-driven feature selection methods such as principal component analysis (PCA) to address this issue [67], as we mentioned in Section 1, It would cause serious

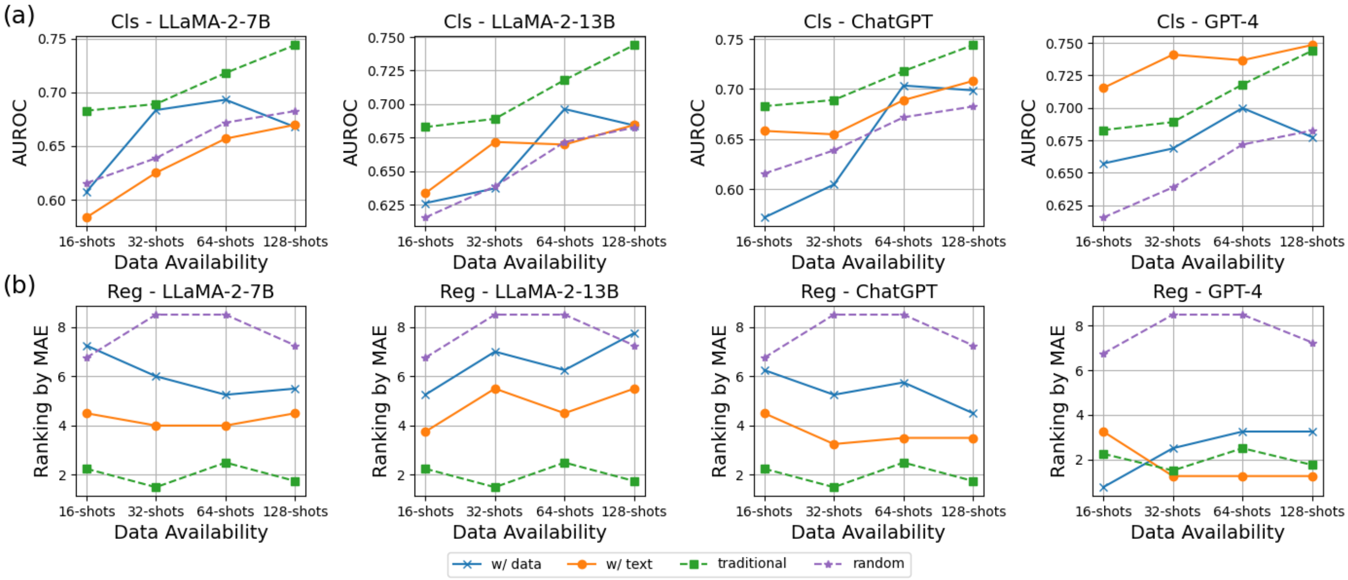


Figure 4: (a) Each feature selection method's results in the classification task, categorized by different LLMs; for each method, we add an error bar to represent its standard variant among various data availabilities. (b) Each feature selection method's results in the regression task, categorized by different LLMs. In each sub-figure, we include the average performance of traditional data-driven methods and the random selection method for comparison.

privacy concerns for both patients and hospitals.

Impressed by the competitive performance and sample-free nature of text-based feature selection with LLMs, here we adopt it in the survival prediction application. In our preliminary experiments, we found LLMs have difficulties in directly understanding the domain-specific feature name (e.g., gene ID). Therefore, we borrow insights from retrieval-augmented generation (RAG) with LLMs [15; 7; 34] and propose **R**etrieval-**A**ugmented **F**eature **S**election **(RAFS)** to efficiently handle these biomedical-specific feature names. Specifically, we retrieve meta information (e.g., official full name, summary and annotation information) about each feature name from the online National Center for Biotechnology Information (NCBI)[5] and provide this information to LLMs as the support document for better feature selection.

5.1 Experiment Settings

We conduct experiments using the Lung Adenocarcinoma (LUAD) dataset in The Cancer Genome Atlas (TCGA) benchmark [56]. Akin to [67], we use clinical indicators and gene expression as the full feature set and fix the feature selection ratio to be 30%. We use PriorityLasso [29] as our machine learning backbone and report three metrics: Antolini's Concordance (Antolini's C) [56], Integrated Brier score (IBS) [17] and D-Calibration (D-CAL) [22], all of which are commonly-used metrics for survival prediction.

5.2 Result Analysis

As the results show in Table 3, we find that even training the model on a randomly selected subset yields slightly better performance than training on the full feature set. This implies the huge volume of features in TCGA-LUAD negatively impacts model performance, highlighting the importance of feature selection. Moreover, we notice feature selection with our RAFS leads to significant performance improvements and consistently outperforms the random selection baseline. These findings suggest that RAFS is an effective approach for handling privacy-sensitive and large-scale biomedical datasets.

	Antolini's C↑	IBS↓	D-CAL↓
PriorityLasso	0.6306	0.1863	1.8518
w/ random	0.6516	0.1833	2.0255
w/ RAFS	**0.6566**	**0.1830**	**1.7666**

Table 3: Experiment results in TCGA-LUAD. We add random selection as the baseline to compare our RAFS with.

6. OUTLOOK

In this section, we discuss potential opportunities for LLMs in feature selection, aiming to provide guidelines and hints for future works.

Synergy of LLMs-based and traditional feature selection. As we discuss in Sections 1 and 4.2, text-based feature selection with LLMs is competitive and resource-efficient compared with traditional feature selection methods. However, each approach relies on different sources of information—specific samples or context descriptions to perform feature selection. This diversity in information utilization makes them complementary. It would be valuable to explore how to combine text-based and traditional feature selection methods to create more effective and robust feature selection systems across various data availability scenarios. Also, it would be interesting to explore the synergy of text-based and data-driven methods to further enhance LLMs-based feature selection under resource constrains.

Data-driven analysis with Agentic LLMs. In Section 4.2, we conclude that poor statistical inference capabil-

[5] https://www.ncbi.nlm.nih.gov/

ities in long-sequence input hinder LLMs in data-driven feature selection. While this finding implies the sole adaptation of LLMs may not be enough for performing data-driven feature selection, the introduction of agent-based LLMs should be considered as an alternative [68; 61]. These methods equip LLM with various tools [47; 70; 51] and APIs [48; 41], enabling them to execute actions and plans to solve complex and multi-step problems. However, there are only a few works that focus on the development of agentic LLMs as data engineers and analytics [25; 14; 59], for actively performing various features or data processing with the assistance of statistical tools or software. Research in this direction will be valuable for enhancing and evaluating LLMs from analytical and statistical perspectives.

Foundation models for feature/data engineering. Many recent works have developed various foundation models in many data mining and machine learning fields, such as graph learning [38; 43; 69] and time series prediction [49; 27]. A large foundation model for feature/ data engineering should be able to understand different types of information from the datasets and perform efficient manipulation and processing [9] to prepare appropriate data for downstream models/ applications. Developing such a foundation model would greatly benefit the data mining and machine learning communities by providing a unified, easy-to-use interface for complex data processing tasks.

7. CONCLUSION

In this study, we explore feature selection methods based on LLMs from a data-centric perspective. We categorize existing LLM-based feature selection approaches into two main types: data-driven, which relies on statistical inference from specific samples, and text-based, which utilizes the extensive knowledge of LLMs for semantic association. Our experiments and analyses reveal that text-based feature selection with LLMs outperforms data-driven methods in terms of effectiveness, stability, and robustness. Based on these findings, we introduce a Retrieval-Augmented Feature Selection (RAFS) method designed to manage large volumes of domain-specific feature candidates in the context of cancer survival time prediction. Additionally, we provide a comprehensive analysis of the current challenges and potential opportunities at the intersection of LLMs and feature selection/engineering in Section 6, aiming to offer insights and guidance for future research in this area.

Acknowledgments

The material in this presentation is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number, 17STQAC00001-08-00, the U.S. Office of Naval Research (ONR) under grant N00014-21-1-4002, and the National Science Foundation (NSF) under grants IIS-2229461. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security and the National Science Foundation.

8. REFERENCES

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] A. Asuncion, D. Newman, et al. Uci machine learning repository, 2007.

[3] A. Beigi, B. Jiang, D. Li, T. Kumarage, Z. Tan, P. Shaeri, and H. Liu. Lrq-fact: Llm-generated relevant questions for multimodal fact-checking. *arXiv preprint arXiv:2410.04616*, 2024.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers & electrical engineering*, 40(1):16–28, 2014.

[6] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2023.

[7] J. Chen, H. Lin, X. Han, and L. Sun. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762, 2024.

[8] K. Choi, C. Cundy, S. Srivastava, and S. Ermon. Lmpriors: Pre-trained language models as task-specific priors. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

[9] L. Cui, H. Li, K. Chen, L. Shou, and G. Chen. Tabular data augmentation for machine learning: Progress and prospects of embracing generative ai. *arXiv preprint arXiv:2407.21523*, 2024.

[10] M. Dash and H. Liu. Feature selection for classification. *Intelligent data analysis*, 1(1-4):131–156, 1997.

[11] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

[12] Z. Dong, T. Tang, J. Li, W. X. Zhao, and J.-R. Wen. Bamboo: A comprehensive benchmark for evaluating long text modeling capacities of large language models. *arXiv preprint arXiv:2309.13345*, 2023.

[13] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–451, 2004.

[14] X. Fang, W. Xu, F. A. Tan, J. Zhang, Z. Hu, Y. J. Qi, S. Nickleach, D. Socolinsky, S. Sengamedu, C. Faloutsos, et al. Large language models (llms) on tabular data: Prediction, generation, and understanding-a survey. *Transactions on Machine Learning Research*, 2024.

[15] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

[16] S. Golovenkin, V. Shulman, D. Rossiev, P. Shesternya, S. Nikulina, Y. Orlova, and V. Voino-Yasenetsky. Myocardial infarction complications. UCI Machine Learning Repository, 2020. DOI: https://doi.org/10.24432/C53P5M.

[17] E. Graf, C. Schmoor, W. Sauerbrei, and M. Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, 18(17-18):2529–2545, 1999.

[18] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.

[19] Q. Gu, Z. Li, and J. Han. Generalized fisher score for feature selection. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 266–273, 2011.

[20] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[21] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46:389–422, 2002.

[22] H. Haider, B. Hoehn, S. Davis, and R. Greiner. Effective ways to build and evaluate individual survival distributions. *Journal of Machine Learning Research*, 21(85):1–63, 2020.

[23] S. Han, J. Yoon, S. O. Arik, and T. Pfister. Large language models can automatically engineer features for few-shot tabular learning. In *Forty-first International Conference on Machine Learning*, 2024.

[24] P. E. Hart, D. G. Stork, R. O. Duda, et al. *Pattern classification*. Wiley Hoboken, 2000.

[25] S. Hong, Y. Lin, B. Liu, B. Wu, D. Li, J. Chen, J. Zhang, J. Wang, L. Zhang, M. Zhuge, et al. Data interpreter: An llm agent for data science. *arXiv preprint arXiv:2402.18679*, 2024.

[26] D. P. Jeong, Z. C. Lipton, and P. Ravikumar. Llm-select: Feature selection with large language models. *arXiv preprint arXiv:2407.02694*, 2024.

[27] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

[28] A. Kadra, M. Lindauer, F. Hutter, and J. Grabocka. Well-tuned simple nets excel on tabular datasets. *Advances in neural information processing systems*, 34:23928–23941, 2021.

[29] S. Klau, V. Jurinovic, R. Hornung, T. Herold, and A.-L. Boulesteix. Priority-lasso: a simple hierarchical approach to the prediction of clinical outcome using multi-omics data. *BMC bioinformatics*, 19:1–14, 2018.

[30] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[31] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. Nowe. A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM transactions on computational biology and bioinformatics*, 9(4):1106–1119, 2012.

[32] D. D. Lewis. Feature selection and feature extraction for text categorization. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

[33] D. Li, Z. Tan, T. Chen, and H. Liu. Contextualization distillation from large language model for knowledge graph completion. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 458–477, 2024.

[34] D. Li, S. Yang, Z. Tan, J. Y. Baik, S. Yun, J. Lee, A. Chacko, B. Hou, D. Duong-Tran, Y. Ding, et al. Dalk: Dynamic co-augmentation of llms and kg to answer alzheimer's disease questions with scientific literature. *arXiv preprint arXiv:2405.04819*, 2024.

[35] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6):1–45, 2017.

[36] Y. Li, A. Dao, W. Bao, Z. Tan, T. Chen, H. Liu, and Y. Kong. Facial affective behavior analysis with instruction tuning. *arXiv preprint arXiv:2404.05052*, 2024.

[37] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

[38] J. Liu, C. Yang, Z. Lu, J. Chen, Y. Li, M. Zhang, T. Bai, Y. Fang, L. Sun, P. S. Yu, et al. Towards graph foundation models: A survey and beyond. *arXiv preprint arXiv:2310.11829*, 2023.

[39] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

[40] S. Liu, F. Lvu, X. Liu, et al. Ice-search: A language model-driven feature selection approach. *arXiv preprint arXiv:2402.18609*, 2024.

[41] X. Liu, Z. Li, P. Li, S. Xia, X. Cui, L. Huang, H. Huang, W. Deng, and Z. He. Mmfakebench: A mixed-source multimodal misinformation detection benchmark for lvlms. *arXiv preprint arXiv:2406.08772*, 2024.

[42] S. Luo and Z. Chen. Sequential lasso cum ebic for feature selection with ultra-high dimensional feature space. *Journal of the American Statistical Association*, 109(507):1229–1240, 2014.

[43] H. Mao, Z. Chen, W. Tang, J. Zhao, Y. Ma, T. Zhao, N. Shah, M. Galkin, and J. Tang. Graph foundation models. *arXiv preprint arXiv:2402.02216*, 2024.

[44] S. Moro, P. Cortez, and P. Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

[45] OpenAI. Introducing chatgpt. *OpenAI*, 2022.

[46] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[47] B. Paranjape, S. Lundberg, S. Singh, H. Hajishirzi, L. Zettlemoyer, and M. T. Ribeiro. Art: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014*, 2023.

[48] S. G. Patil, T. Zhang, X. Wang, and J. E. Gonzalez. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*, 2023.

[49] K. Rasul, A. Ashok, A. R. Williams, A. Khorasani, G. Adamopoulos, R. Bhagwatkar, M. Biloš, H. Ghonia, N. V. Hassen, A. Schneider, et al. Lag-llama: Towards foundation models for time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.

[50] M. Redmond. Communities and Crime. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C53W3X.

[51] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2024.

[52] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(5), 2012.

[53] Z. Tan, A. Beigi, S. Wang, R. Guo, A. Bhattacharjee, B. Jiang, M. Karami, J. Li, L. Cheng, and H. Liu. Large language models for data annotation: A survey. *arXiv preprint arXiv:2402.13446*, 2024.

[54] Z. Tan, J. Peng, T. Chen, and H. Liu. Tuning-free accountable intervention for llm deployment–a metacognitive approach. *arXiv preprint arXiv:2403.05636*, 2024.

[55] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[56] K. Tomczak, P. Czerwińska, and M. Wiznerowicz. Review the cancer genome atlas (tcga): an immeasurable source of knowledge. *Contemporary Oncology/Współczesna Onkologia*, 2015(1):68–77, 2015.

[57] Y. Tong, D. Li, S. Wang, Y. Wang, F. Teng, and J. Shang. Can llms learn from previous mistakes? investigating llms' errors to boost for reasoning. *arXiv preprint arXiv:2403.20046*, 2024.

[58] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[59] P. Trirat, W. Jeong, and S. J. Hwang. Automl-agent: A multi-agent llm framework for full-pipeline automl. *arXiv preprint arXiv:2410.02958*, 2024.

[60] S. Wadhwa, S. Amir, and B. C. Wallace. Revisiting relation extraction in the era of large language models. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2023, page 15566. NIH Public Access, 2023.

[61] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

[62] S. Wang, Z. Tan, R. Guo, and J. Li. Noise-robust fine-tuning of pretrained language models via external guidance. *arXiv preprint arXiv:2311.01108*, 2023.

[63] X. Wang, Z. Chen, H. Wang, Z. Li, W. Guo, et al. Large language model enhanced knowledge representation learning: A survey. *arXiv preprint arXiv:2407.00936*, 2024.

[64] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[65] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[66] D. Wissel, N. Janakarajan, A. Grover, E. Toniato, M. R. Martínez, and V. Boeva. Survboard: standardised benchmarking for multi-omics cancer survival models. *bioRxiv*, pages 2022–11, 2022.

[67] D. Wissel, D. Rowson, and V. Boeva. Systematic comparison of multi-omics survival models reveals a widespread lack of noise resistance. *Cell Reports Methods*, 3(4), 2023.

[68] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.

[69] L. Xia, B. Kao, and C. Huang. Opengraph: Towards open graph foundation models. *arXiv preprint arXiv:2403.01121*, 2024.

[70] R. Yang, L. Song, Y. Li, S. Zhao, Y. Ge, X. Li, and Y. Shan. Gpt4tools: Teaching large language model to use tools via self-instruction. *Advances in Neural Information Processing Systems*, 36, 2024.

[71] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67, 2006.

[72] T. Zhang, T. Zhu, P. Xiong, H. Huo, Z. Tari, and W. Zhou. Correlated differential privacy: Feature selection in machine learning. *IEEE Transactions on Industrial Informatics*, 16(3):2115–2124, 2019.

APPENDIX

A. DETAILED INSTRUCTION

/* Main System Prompt */
For the given feature, your task is to provide a feature importance score (between 0 and 1; larger value indicates greater importance).

/* Specific Sample Vlaues */
Here are some data points in the format of (feature value, target value), please refer to this to determine how informative the feature is in predicting the target value:
(<0, no)
(no checking, no)
(<0, no)
(<0, no)
(0<=X<200, no)
(<0, no)
(>=200, no)
(<0, no)
(no checking, yes)
(no checking, yes)
(0<=X<200, yes)
(0<=X<200, yes)
(no checking, yes)
(0<=X<200, yes)
(0<=X<200, yes)
(<0, yes)

/* Output Format Instruction */
Here is an example:
‘‘‘
Question: What is the importance score for the given feature
Answer: The importance score is 0.9
‘‘‘

/* Main User Prompt*/
Question: What is the importance score for the given feature
Answer: The importance score is

Table 4: Detailed instruction for data-driven method in Credit-g dataset.

/* Dataset-specific Context */
Context: Using data collected at a German bank, we wish to build a machine learning model that can accurately predict whether a client carries high or low credit risk (target variable). The dataset contains a total of 20 features (e.g., credit history, savings account status). Prior to training the model, we first want to identify a subset of the 20 features that are most important for reliable prediction of the target variable.

/* Main System Prompt */
For each feature input by the user, your task is to provide a feature importance score (between 0 and 1; larger value indicates greater importance) for predicting whether an individual carries high credit risk and a reasoning behind how the importance score was assigned.

/* Output Format Instructions */
The output should be formatted as a JSON instance that conforms to the JSON schema below.

As an example, for the schema ”properties”: ”foo”: ”title”: ”Foo”, ”description”: ”a list of strings”, ”type”: ”array”, ”items”: ”type”: ”string”, ”required”: [”foo”] the object ”foo”: [”bar”, ”baz”] is a well formatted instance of the schema. The object ”properties”: ”foo”: [”bar”, ”baz”] is not well-formatted.

Here is the output schema:
‘‘‘
{”description”: ”Langchain Pydantic output parsing structure.”, ”properties”: {”reasoning”: {”title”: ”Reasoning”, ”description”: ”Logical reasoning behind feature importance score”, ”type”: ”string”}, ”score”: {”title”: ”Score”, ”description”: ”Feature importance score”, ”type”: ”number”}}, ”required”: [”score”]}
‘‘‘
/* Demonstration */
Here is an example output:
-Variable: Installment rate in percentage of disposable income
{”reasoning”: ”The installment rate as apercentage of disposable incomeprovides insight intoa person’s financial responsibility and capability.This percentage can be seen as a measure of how much of a person’s available income is committed to repaying their debts. If this rate is high, it might indicate that the person is taking more debt than they can comfortably repay and may hint a talack off inancial responsibility, implyinghigher credit risk. If this rate is low, it likely indicates that the person can manage their current financial obligations comfortably, implying lowercredit risk. Thus, the score is 0.9.”, ”score”: 0.9}

/*Main User Prompt*/
Provide a score and reasoning for ”Status of existing checking account, in Deutsche Mark.” formatted according to the output schema above:

Table 5: Detailed instruction for text-based method in Credit-g dataset.

Causal inference under limited outcome observability: A case study with Pinterest Conversion Lift

Min Kyoung Kang
Pinterest, Inc.
1099 Stewart St, Seattle
WA, USA
mkang@pinterest.com

ABSTRACT

This paper compares the performance of several established causal inference estimators in measuring conversion related metrics for advertising measurement applications. Conversion lift measurement in advertising industry presents unique challenges due to complex data collection process, potential data losses, and complex customer behaviors leading up to conversion. Case studies with both simulated and real-world data demonstrated that doubly robust estimators outperform regression adjustment estimators in variance reduction for ad measurement use-cases. To further understand the results, we examine the impact of data loss on variance reduction by the estimators and find that the relationship between data loss and variance reduction performance varies by the estimators. Doubly robust estimators could effectively manage complex relationships introduced by data loss, maintaining superior performance over the difference-in-means and regression adjustment estimator in terms of precision under various circumstances. We provide computational cost perspectives as practical considerations for implementing doubly robust estimators in advertising measurement business solutions.

1 Introduction

Accurately measuring the impact of placing advertisement in online platform industry, often referred to as advertising measurement, is a challenging but one of the most important tasks for customer behavior understanding. In particular, measurement reporting for conversion count and volume metrics provides critical guidance on business decisions for optimizing marketing directions based on user understanding. Such measurement process encompasses multiple phases of execution including holdout experiments, data collection, and inferential analysis. Each of these stages has potential to introduce unwanted uncertainty into the data, thereby introducing challenges for the inference [12; 2]. Holdout experiments can suffer from compliance issues meaning that customers may not actually impress the ads even if they are assigned to treatment group. Customer behaviors related to conversions are inherently complex and sparse since in general customers convert occasionally after a number of non-linear interactions. For privacy concerns and data sharing agreement limitations, data loss can occur limiting the observability of outcome metrics. These factors increase the complexity of treatment effect estimation and can potentially decrease the power of experiments. Extending duration of the holdout experiments to account for power incurs opportunity cost for advertisers, as holdout experiments withhold advertisements to control users, limiting the reach to high-potential customers within campaign period.

This work aligns with a number of literature and researches on ways to improve the power and sensitivity of controlled randomized experiments [10; 3; 5; 9; 15]. Such effort is commonly referred as variance reduction techniques, which leverage covariates that explain the variability unrelated to treatment from outcomes to reduce variance of the outcome metrics of interest. This is the area of active investigation and interests as it can decrease the costs of running experiments and extract insights from otherwise inconclusive experimental results. However, there is limited research on variance reduction techniques under the context of advertising measurement with causal inference estimators. Literature in advertising measurement mainly discusses causal inference approach to measure impact of advertising with observational data [4; 7; 13]. We apply causal estimators to holdout experimental settings to evaluate their performance in terms of precision within the unique context of the ad measurement industry, which includes data loss. This work extends existing variance reduction causal inference literature into advertising measurement domain by exploring the opportunities at the intersection of variance reduction in ad measurement research. It aims to inform practitioners in advertising industry who want to improve the sensitivity of their measurement reporting with practical considerations.

The remainder of the paper are organized as follows. Section 2 introduces four established estimators for inferential analysis on holdout experiments in ad measurement use-cases. In section 3, we compare the performance of the estimators with case studies and performs empirical analysis with both real-world and simulated data. We conclude the paper with the discussion on production adoption and business impact considerations while implementing the method in practice.

2 Estimators for advertising measurement

This section introduces the formal definitions of estimators that will be compared in case studies to empirically analyze their performance in the context of conversion advertising measurement. A randomized controlled trial (RCT), also referred to as online randomized experiment, is an industry-standard measure for measuring incrementality of advertising without confoundings. To formally denote the experimental settings, we denote a treatment as T, outcome as Y, covariate as X, and one instance of experimental data points with i. Historical data we observe provide scientists with a set of data for each instance (T_i, Y_i, X_i), where X is a set of covariate vectors that are presumed to be closely related with Y according to empirical evidence. Under controlled randomized experimentation, $(Y_i(0),\ Y_i(1)) \perp T_i$ holds. Sometimes experiment experiences opportunistic imbalance due to experiment quality issues, but we can assume $(Y_i(0),\ Y_i(1)) \perp T_i|X_i$.

2.1 Difference-in-Means estimator

The difference-in-means estimator (DIM) is a traditional statistical approach that calculate delta between average outcome of treatment and control groups. This provides information on the significant differences between the teams of two populations.

$$\text{ATE}_{dim} = \overline{Y}_{tr} - \overline{Y}_{ctl} \tag{1}$$

, where $\overline{Y}_{group}$ is for users that belong to a particular group (treatment or control) and defined as $\frac{1}{n_{group}} \sum_{j \in group} y_j$ for outcome of interest y for individuals from control and treatment groups.

2.2 Regression adjustment estimator

Leveraging unit-level covariates in post-experiment inferential stage is known to enhance the precision and efficiency of causal estimates. A traditional method for adjusting treatment effect with covariates is ordinary least square (OLS) regression adjustment (RA). This approach has been studied in various literatures and known for its ability to asymptotically improve the precision of estimators [15; 11; 9; 14; 17]. This work examines two model specifications for regression adjustment estimators and the only difference between the two models is the inclusion of interaction terms between the treatment and covariates.

$$Y_i = \alpha + \tau T_i + \beta X_i + \epsilon_i \tag{2}$$

$$Y_i = \alpha + \tau T_i + \beta X_i + \gamma(T_i \cdot X_i) + \epsilon_i \tag{3}$$

The literature demonstrated that (3) yields better statistical properties [15]. For both of model specifications, the average treatment effect is estimated by solving the regular OLS optimization process to determine model parameter τ.

2.3 Doubly Robust Estimator

Doubly robust (DR) estimator is a causal inference approach to estimate the treatment effect in a doubly robust manner [6; 1]. The robustness comes from incorporating two modeling approaches in calculating estimators, which are propensity score model and outcome model. DR estimators have desirable statistical properties of consistency and efficiency. As long as either of the propensity score model or the outcome model is correctly specified, the doubly robust estimator yields a consistent estimator. Thus, subtle misspecification of either of the outcome and propensity model does not affect the treatment effect estimation. Such double (or dual) robustness characteristics increase the chances that the estimator has minimal population risk in practical applications. When both models are correctly specified, the doubly robust estimator can account for available covariates' into the model and achieves the resulting lowest possible variance. As the methodology can accommodate various machine learning models with regularization, it can handle high-dimensional covariates with both nonparametric and parametric functions to account for complex relationships among various factors. To formally introduce DR estimator, the potential outcome under treatment assignment of t for unit i is formulated with outcome model $f_t(x) = E[Y(T = t)|X = x]$ and propensity score model $p(t, x) = E[T = t|X = x]$ as

$$\widehat{Y}_i(t) = f_t(x) + \frac{(Y_i - f_t(x))}{p(t, x)} \mathbf{1}(T = t) \tag{4}$$

With the provided potential outcome, treatment effect function ϕ can be estimated with the following minimization process, which can be customized to account for effect heterogeneity as needed.

$$argmin_\phi \sum_i (\widehat{Y}_i(1) - \widehat{Y}_i(0) - \phi(X))^2 \tag{5}$$

To obtain ϕ through this optimization process, nuisance parameters such as f (outcome model) and p (propensity score model) are estimated from the data using machine learning techniques. For the analysis of empirical results, we selected random forest models for both f and p, which is tuned using 4-fold cross validation processes on the entire dataset to effectively capture sparse conversion data. As advertisers are most interested in average impact of the advertisement, we estimate average treatment effect assuming the effect homogeneity across population. Thus, for the analysis for the following case studies, $\phi(X)$ is reduced to a simple estimation with $\alpha + \tau T_i$ that measures homogeneous treatment effects in this analysis.

To maintain the consistency of the inference process across various estimators, we construct the analytical confidence interval for all estimators with OLS parameter τ, coefficient of treatment assignment variable T_i. Based on the asymptotic normality of τ, we construct Wald 95% confidence interval, which is a traditional approach for regression analysis. More detailed explanation of calculating confidence interval from OLS optimization can be found in [16].

3 Empirical results with case studies

This section utilizes causal estimators introduced in previous section under the context of controlled randomized experimentation to understand the variance reduction performance of the estimators introduced. For this analysis, both real-user data from pinterest conversion lift studies and simulated data are used to understand the impact of various environmental factors in variance reduction performance under advertising measurement context. This analysis adopts three evaluation metrics: the first assesses the level of variance reduction, the second measures the coverage probability of the confidence interval when ground truth value is available, and the third quantifies the deviation between true and estimated effects at point-estimate level. The second and third metrics are calculated exclusively when ground truth treatment effects are available, as in the case of simulated data. The level of variance reduction is measured against the relative confidence interval ratio of 3 estimators against of difference in means estimator, which is baseline measurement solutions in industry. All confidence interval is calculated at a 95% of significance level.

$$\mathrm{VR}_{est} = CI_{est}/CI_{DIM} \tag{6}$$

The coverage percentage metrics indicate the proportion of confidence intervals, each obtained from iteration of simulated data, that includes the treatment effect within their lower and upper bound.

$$\mathrm{Cov\%}_{\mathrm{est}} = \frac{\sum_{j=1}^{N} \mathbf{1}(\text{lower bound}_j \leq \tau_j \leq \text{upper bound}_j)}{N} \tag{7}$$

The mean squared error (MSE) metric represents the averaged of squared error, which is the delta between ground truth and calculated treatment effect from the estimators.

$$\mathrm{MSE}_{\mathrm{est}} = \sum_{j=1}^{N} (\hat{\tau_j} - \tau_j)^2 \tag{8}$$

We assess the performance of four estimators ATE_{DIM} (difference in mean), $ATE_{RA(3)}$ (OLS adjustment without interaction), $ATE_{RA(4)}$ (OLS adjustment with interaction), ATE_{DR} (doubly robust estimator) utilizing three performance metrics in the remainder of sections.

3.1 Pinterest conversion lift (PCL) case studies

Pinterest conversion lift (PCL) study quantifies the value of placing advertisements on the Pinterest platform for various lower-funnel performance advertisements. This quantification includes a wide range of customer conversion-related engagement following exposure of performance ads in Pinterest, allowing advertisers to measure and optimize campaign performance. One of the goals of PCL study is to provide advertisers with sufficient statistical power, so that the study delivers maximum value to advertisers with accurate and actionable insights within their campaign budget constraints. This aims to minimize the number of studies yielding inconclusive results attributing from the high variability of outcome metrics. This case study compares the estimators aiming to decrease variability of outcome metrics unrelated to the treatment of the holdout experiment to increase statistical power of the experiment.

Table 1: Quantitative summary of the performance of the estimators for outcome metrics from PCL studies. ('Change in stat sig rate' column displays the difference in statistical significance rate between the DIM estimator and the selected estimator of interest.)

Outcome	Avg %			
	Reduction in Variance (DR)	Reduction in Variance (RA(3))	Reduction in Variance (RA(4))	Change in stat sig rate
Metric 1	7.6%	0.06%	0.06%	3.0%
Metric 2	9.8%	0.19%	0.21%	6.9%
Metric 3	13.1%	0.05%	0.06%	11.2%

To evaluate the variance reduction performance, we compared the confidence interval using variance reduction metrics with difference-in-means estimator as the baseline model; PCL currently utilizes difference in means estimators following the holdout experiment data collection. Treatment effect estimates yielded from various estimators based on selected historical PCL's experimental data. These experiments are randomly selected from delivered studies to advertisers dating back as early as 2023 July. The advertisers selected for this study represent a diverse set of industries, including but not limited to online retail, telecom services, finance, etc. Experiment data is collected from various regions, including North and South America, as well as the EU.

For each of experiment, three types of outcome metrics are considered, which are denoted as metric 1, 2, 3, anonymized to maintain the confidentiality of sensitive internal data. Due to the unkown ground truth treatment effect, we solely focused on comparing variance reduction metrics and did not calculate coverage probability or mean squared error (MSE) metrics for the performance measurement. However, it is important to note that the estimated treatment effects across different estimators are generally consistent, exhibiting minor variations and largely overlapping confidence intervals. The performance metrics indicate relative measures to compare the level of variance reduction relative to difference-in-means estimator. Experimental outcome metric names are anonymized and absolute values of performance metrics are not shown to maintain the confidentiality of sensitive internal data. For the estimation of the treatment effect and confidence interval, we utilized holdout experiment data from historical PCL studies and a number of covariates are collected by summarizing various information of individual user activities on Pinterest platform prior to the experiment start date.

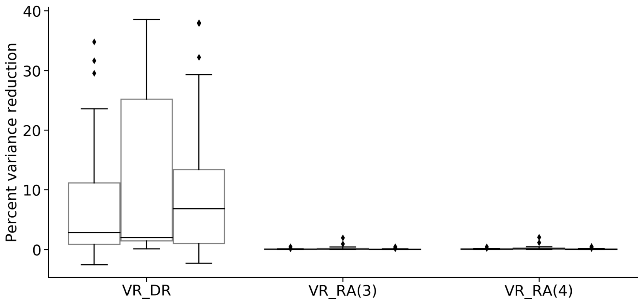


Figure 1: Box plots for variance reduction performance comparisons. This plot compares the percentage of variance reduction across metrics 1, 2, and 3 for the estimators introduced in section 2. Each group of box plots represents the performance of each estimator; within each group, three boxes correspond to metrics 1, 2, and 3, respectively. The vertical axis displays the percentage of reduction in variance, providing a visual representation of the distribution of reduction across selected PCL studies.

For almost all experimental cases selected, DR estimator consistently outperformed regression adjustment estimators in Fig.1 and TABLE 1, demonstrating its effectiveness in accounting for variability orthogonal to the experimental treatment under complex data structures.

For the doubly robust estimator, the percentage reduction in variance is consistently higher across all three metrics. This indicates a reliable performance in reducing variance under various circumstances, leading to the greatest increase in the statistical significance rate. Subtle differences in the level of percentage reduction are explained by metric types; metrics involving volumes tend to have higher variance due to the larger values associated with data. In contrast, metrics such as counts, which involve smaller integer values, typically exhibit lower variance. The doubly robust estimator is more effective at reducing variability in metrics with larger volumes and higher inherent variability compared to those with lower activity and smaller numerical values. Interestingly, regression adjustment was ineffective in reducing variance and increasing the statistical significance rate. These estimators utilize linear models, which may have limitations in capturing the complex nature of user behaviors and advertising conversion data, leading to lower statistical significance in measuring treatment effects. We deep dive into such dynamics with simulated data in the next subsection.

3.2 Simulated data case studies

To assess the performance of the estimators for the purpose of advertising measurement, we generated simulated data with constant and homogeneous treatment effect across units of observation. Treatment assignment is randomized without confounding, while outcomes are influenced by both treatment effect based on assignment status and other randomly generated covariates. For the simulated data, we adopt a data generating process (DGP) in [8] to account for complex non-linear relationship between outcome and various factors as in real experimental set-up. This DGP exhibits varying fluctuation and smoothness across multi-dimensional variables included in the formulation. Each iteration of simulation randomly generates data of size $N = 100,000$ units and this process is repeated $5,000$ times to assess their average performance. To account for unique data collection process in ad measurement scenario, we randomly mask outcome data under the scenario of varying level of data loss. The following information describes the variables used to generate simulated data and to estimate treatment effects:

Table 2: Coverage %, Variance Ratio, and Mean Squared Error for different estimators

	DR	**RA(3)**	**RA(4)**	**DIM**
Cov %	94.72%	94.64%	94.45%	94.52%
VR	15.0	11.5	11.6	1
MSE	0.024	0.025	0.025	0.028

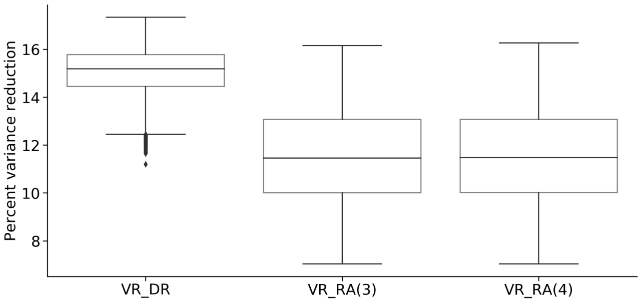


Figure 2: Comparison of percentage variance reduction for various estimators versus difference in means estimator

- Y_i: denotes the outcome variable for the i-th instance. Outcome, which is affected by both four covariates and treatment.

- T_i: denotes the treatment variable for the i-th instance. Under holdout experimental set-up, treatment assignment mechanism is randomized. Thus, treatment is simulated with Bernoulli distribution as $T_i \sim Bernoulli(1/2)$

- X_{ij}: denotes the j-th covariates for the i-th instance. All covariates follow independent and identically distributed normal distributions with mean zero and variance one. We generated 100 covariates and only 4 of them are used for DGP process, while rest of 96 covariates are unrelated to outcome. All of 100 covariates are included in the treatment effect estimation process.

- τ_i: represents the injected treatment effect which we adjust to account for random data loss when calculating the ground truth effect that we aim to estimate in this case study. For each of simulation data generation, $\tau_i \sim \text{Uniform}(0.5, 1.5)$ to validate the model performance under various conditions.

- m_i: represents the observability rate for the outcome Y_i. Given the data loss is common in collecting conversion labels in ad measurement industry, we randomly mask some portion of data at a rate of $1 - m_i$, where $m_i \sim \text{Uniform}(0.7, 0.95)$

- I_i: represents the indicator variable for each unit to decide its outcome loss status. $I_i \sim Bernoulli(m_i)$ and I_i is used to mask outcome Y_i to be zero. $I_i = 0$ induces the scenario of outcome data loss for i-th instance.

- ϵ : represents normally distributed random noise with mean zero and variance one $\sim \mathcal{N}(0, 1)$

With the notations defined, the outcome of interest is formulated to reflect complex relationship adopted from [8]:

$$\begin{aligned} Y_i * I_i \; &= \; \tau T_i + \\ & \exp\left(\sin\left(0.9 \cdot (X_{i1} + 0.48)^{10}\right)\right) + \\ & (X_{i2} \cdot X_{i3}) + X_{i4} + \epsilon \end{aligned} \tag{9}$$

Fig.2 compares the percentage of variance reduction for various estimators compared to difference-in-means estimator. The vertical axis displays the percentage of reduction in variance, informing the distribution of reduction achieved by each estimator for simulated data under various data loss. The simulation study confirmed again that DR estimator consistently outperforms regression adjustment estimator for variance reduction. Quantitative comparisons of the performance of estimators are in TABLE 2. DR estimator also enhances the precision of metrics decreasing mean squared errors.

We further deep dive into the variance reduction performance of each estimator by the observability rate, which mimic various data loss scenario unique to ad measurement data. The purpose is to understand the relation between data loss and variance reduction performance to identify more appropriate estimators under ad measurement use-cases. The simulation analysis suggests that the regression adjustment models' variance reduction performance decreases as data loss increases. This results explain the PCL case study where we found that regression adjustment analysis was not able to further reduce the variance compared to DIM estimator.

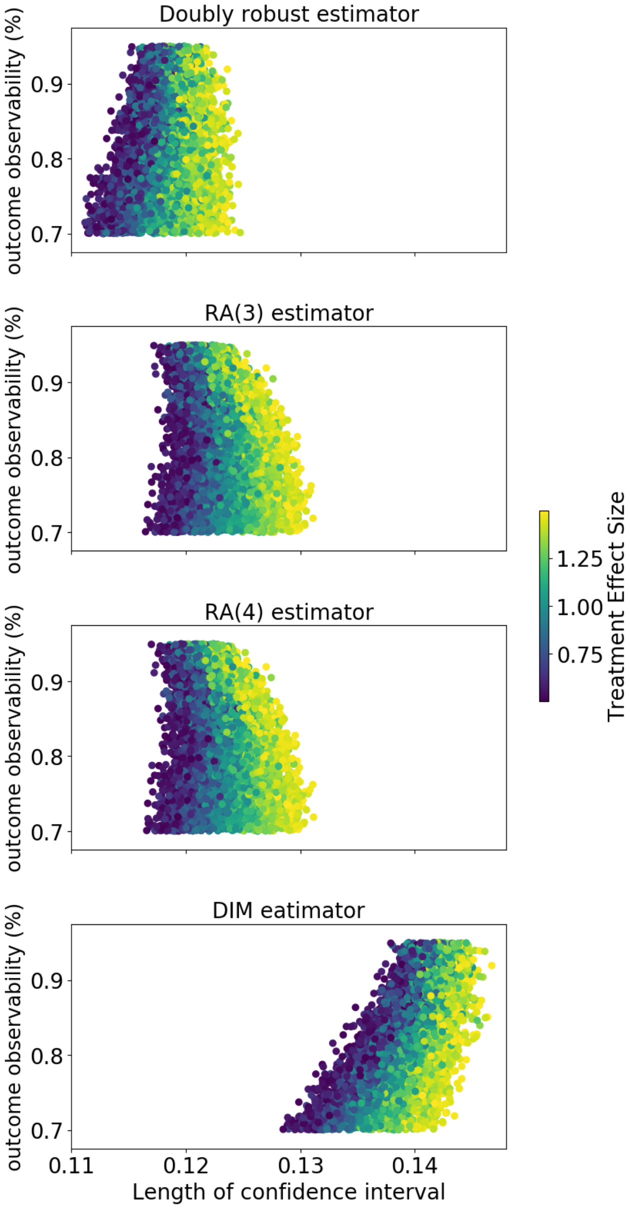


Figure 3: Scatter plots that compare the confidence intervals across estimators by outcome observability and treatment effect size

Fig.3 compares the confidence interval calculated from four different estimators by the various level of outcome observability and treatment effect size. Scatter plots from four estimators show that as outcome observability grows, regression adjustment based estimator's confidence interval starting to overlap with DIM estimator's, while doubly robust estimator maintains its relative advantage.

4 Conclusions

In this paper, we compared several established estimators in the causal inference literature, particularly focusing on their performance for advertising measurement purposes. Advertising measurement presents unique challenges due to the complexities involved in data collection, subsequent data losses, and the intricate relationships across various factors within the data. Our case studies, which included both simulated and real-world data, demonstrated that doubly robust estimators outperform regression adjustment estimators in variance reduction while maintaining coverage probability and reducing the error in point estimates. We also identified the impact of data loss on variance reduction through simulation studies. By examining the relationship between the proportion of data loss and relative variance reduction performance, we observed that as the data loss percentage increases, the regression adjustment model decrease its variance reduction performance trending toward the level of difference in means estimator, while the doubly robust estimator maintains its superiority in variance reduction. Machine learning models used in doubly robust estimators could better handle unexpected intricate relationships introduced into the data due to data loss, compared to regression adjustment-based estimators. Doubly robust estimators are also well-known for their ability to provide robust estimates under mis-specifications of either outcome or propensity score models. In complex non-linear datasets with stochastic factors, the properties of doubly robust estimator improves the precision of inference results.

The inference process of online experiments often involves massive data volumes and training machine learning models on sizeable data can pose challenges due to computational costs. While cross-fitting could further reduce bias, it may introduce challenges due to massive data size, operational costs, and data sparsity. By adopting cross-validation process in model training, we prevent over-fitting of ML models utilized for outcomes and propensity score models, thereby minimizing potential bias issues due to over-fitting. To further mitigate cost related challenges, one can also consider a two-tiered services to leverage high variance reduction solutions when necessary. With this approach, we offer the practical perspectives of the measurement solution, balancing variance reduction performance against costs and scalability considerations.

5 REFERENCES

[1] H. Bang and J. M. Robins. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973, 2005.

[2] J. Barajas, N. Bhamidipati, and J. G. Shanahan. Online advertising incrementality testing: practical lessons and emerging challenges. In *Proceedings of the 30th ACM*

International Conference on Information & Knowledge Management, pages 4838–4841, 2021.

[3] S. Baweja, N. Pokharna, A. Ustimenko, and O. Jeunen. Variance reduction in ratio metrics for efficient online experiments. In *European Conference on Information Retrieval*, pages 292–297. Springer, 2024.

[4] D. Chan, R. Ge, O. Gershony, T. Hesterberg, and D. Lambert. Evaluating online ad campaigns in a pipeline: causal models at scale. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 7–16, 2010.

[5] A. Deng, Y. Xu, R. Kohavi, and T. Walker. Improving the sensitivity of online controlled experiments by utilizing pre-experiment data. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 123–132, 2013.

[6] M. J. Funk, D. Westreich, C. Wiesen, T. Stürmer, M. A. Brookhart, and M. Davidian. Doubly robust estimation of causal effects. *American journal of epidemiology*, 173(7):761–767, 2011.

[7] B. R. Gordon, F. Zettelmeyer, N. Bhargava, and D. Chapsky. A comparison of approaches to advertising measurement: Evidence from big field experiments at facebook. *Marketing Science*, 38(2):193–225, 2019.

[8] R. B. Gramacy and H. K. Lee. Adaptive design and analysis of supercomputer experiments. *Technometrics*, 51(2):130–145, 2009.

[9] K. Guo and G. Basse. The generalized oaxaca-blinder estimator. *Journal of the American Statistical Association*, 118(541):524–536, 2023.

[10] Y. Guo, D. Coey, M. Konutgan, W. Li, C. Schoener, and M. Goldman. Machine learning for variance reduction in online experiments. *Advances in Neural Information Processing Systems*, 34:8637–8648, 2021.

[11] G. W. Imbens and J. M. Wooldridge. Recent developments in the econometrics of program evaluation. *Journal of economic literature*, 47(1):5–86, 2009.

[12] G. A. Johnson, R. A. Lewis, and E. I. Nubbemeyer. Ghost ads: Improving the economics of measuring online ad effectiveness. *Journal of Marketing Research*, 54(6):867–884, 2017.

[13] R. A. Lewis, J. M. Rao, and D. H. Reiley. Here, there, and everywhere: correlated online behaviors can lead to overestimates of the effects of advertising. In *Proceedings of the 20th international conference on World wide web*, pages 157–166, 2011.

[14] X. Li and P. Ding. Rerandomization and regression adjustment. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(1):241–268, 2020.

[15] W. Lin. Agnostic notes on regression adjustments to experimental data: Reexamining freedman's critique. 2013.

[16] J. Neter, M. H. Kutner, C. J. Nachtsheim, W. Wasserman, et al. Applied linear statistical models. 1996.

[17] A. A. Tsiatis, M. Davidian, M. Zhang, and X. Lu. Covariate adjustment for two-sample treatment comparisons in randomized clinical trials: a principled yet flexible approach. *Statistics in medicine*, 27(23):4658–4677, 2008.

DiffusionShield: A Watermark for Data Copyright Protection against Generative Diffusion Models

Yingqian Cui[1]*, Jie Ren[1]*, Han Xu[2], Pengfei He[1], Hui Liu[1],
Lichao Sun[3], Yue Xing[1], Jiliang Tang[1]
[1]Michigan State University [2]The University of Arizona
[3]Lehigh University
{cuiyingq, renjie3, hepengf1, liuhui7, xingyue1,
tangjili}@msu.edu xuhan2@arizona.edu lis221@lehigh.edu

ABSTRACT

Recently, Generative Diffusion Models (GDMs) have shown remarkable abilities in learning and generating images, fostering a large community of GDMs. However, the unrestricted proliferation has raised serious concerns on copyright issues. For example, artists become concerned that GDMs could effortlessly replicate their unique artworks without permission. In response to these challenges, we introduce a novel watermark scheme, DiffusionShield, against GDMs. It protects images from infringement by encoding the ownership message into an imperceptible watermark and injecting it into images. This watermark can be easily learned by GDMs and will be reproduced in generated images. By detecting the watermark in generated images, the infringement can be exposed with evidence. Benefiting from the uniformity of the watermarks and the joint optimization method, DiffusionShield ensures low distortion of the original image, high watermark detection performance, and lengthy encoded messages. We conduct rigorous and comprehensive experiments to show its effectiveness in defending against infringement by GDMs and its superiority over traditional watermark methods.

1. INTRODUCTION

Generative diffusion models (GDMs), such as Denoising Diffusion Probabilistic Models (DDPM) [11] have shown their great potential in generating high-quality images. This has also led to the growth of more advanced techniques, such as DALL·E2 [23], Stable Diffusion [24], and ControlNet [38]. In general, a GDM learns the distribution of a set of collected images, and can generate images that follow the learned distribution. As these techniques become increasingly popular, concerns have arisen regarding the copyright protection of creative works shared on the Internet. For instance, a fashion company may invest significant resources in designing a new fashion. After the company posts the pictures of this fashion to the public for browsing, an unauthorized entity can train their GDMs to mimic its style and appearance, generating similar images and producing products. This infringement highlights the pressing need for copyright protection mechanisms.

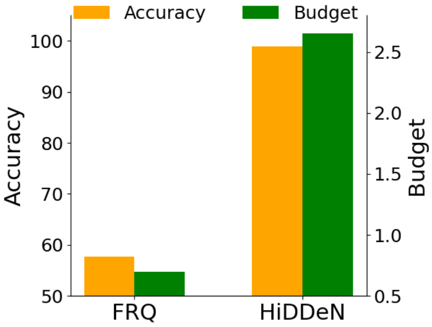


Figure 1: Watermark detection accuracy (%) on GDM-generated images and the corresponding budget (l_2 norm) of watermarks.

To provide protection for creative works, watermark techniques such as [5, 22, 44, 18, 35] are often applied, which aim to inject (invisible) watermarks into images and then detect them to track the malicious copy and accuse the infringement. However, directly applying these existing methods to GDMs still faces tremendous challenges. Indeed, since existing watermark methods have not specifically been designed for GDMs, they might be hard to learn by GDMs and could disappear in the generated images. Then the infringement may not be effectively verified and accused.

An empirical evidence can be found in Figure 1. We train two popular GDMs on a CIFAR10 dataset whose samples are watermarked by two representative watermark methods [18, 44], and we try to detect the watermarks in the GDM-generated images. The result demonstrates that the watermarks from these methods are either hardly learned and reproduced by GDM (e.g., FRQ [18]), or require a very large budget (the extent of image distortion) to partially maintain the watermarks (e.g., HiDDeN [44]). Therefore, dedicated efforts are still greatly desired to developing the watermark technique tailored for GDMs.

In this work, we argue that one critical factor that causes the inefficacy of these existing watermark techniques is the inconsistency of watermark patterns on different data samples. In methods such as [18, 44], the watermark in each image from one owner is distinct. Thus, GDMs can hardly learn the distribution of watermarks and reproduce them in the generated samples. To address this challenge, we propose **DiffusionShield** which aims to enhance the "*pattern*

*Equal contribution

uniformity" (Section 3.2) of the watermarks to make them consistent across different images. We first empirically show that watermarks with pattern uniformity are easy to be reproduced by GDMs in Section 3.2. Then, we provide corresponding theoretic analysis in two examples to demonstrate that the watermarks with pattern uniformity will be learned prior to other features in Section 3.5. The theoretical evidence further suggests that if unauthorized GDMs attempt to learn from the watermarked images, they are likely to learn the watermarks before the original data distribution.

Leveraging pattern uniformity, DiffusionShield designs a blockwise strategy to divide the watermarks into a sequence of basic patches, and a user has a specific sequence of basic patches which forms a watermark applied on all his/her images and encodes the copyright message. The watermark will repeatedly appear in the training set of GDMs, and thus makes it reproducible and detectable. In the case of multiple users, each user will have his/her own watermark pattern based on the encoded message. Furthermore, DiffusionShield introduces a joint optimization method for basic patches and watermark detectors to enhance each other, which achieves a smaller budget and higher accuracy. In addition, once the watermarks are obtained, DiffusionShield does not require re-training when there is an influx of new users and images, indicating its flexibility to accommodate multiple users. In summary, with the enhanced pattern uniformity in blockwise strategy and joint optimization, we can successfully secure the data copyright against infringement by GDMs.

2. RELATED WORK

Generative Diffusion Models. Recently, GDMs have made significant strides. A breakthrough in GDMs is achieved by DDPM [19], which demonstrates great superiority in generating high-quality images. The work of [12] further advances the field by eliminating the need for classifiers in the training process. [27] presents Denoising Diffusion Implicit Models (DDIMs), a variant of GDMs with improved efficiency in sampling. Besides, techniques such as [24] achieve high-resolution image synthesis and text-to-image synthesis. These advancements underscore the growing popularity and efficacy of GDM-based techniques.

To train GDMs, many existing methods rely on collecting a significant amount of training data from public resources [7, 34, 9]. However, there is a concern that if a GDM is trained on copyrighted material and produces outputs similar to the original copyrighted works, it could potentially infringe on the copyright owner's rights. This issue has already garnered public attention [30], and our paper focuses on mitigating this risk by employing a watermarking technique to detect copyright infringements.

Image Watermarking. Image watermarking involves embedding invisible information into the carrier images and is commonly used to identify ownership of the copyright. Traditional watermarking techniques include spatial domain methods and frequency domain methods [5, 18, 26]. These techniques embed watermark information by modifying the pixel values [5], frequency coefficients [18], or both [26, 14]. Recently, various digital watermarking approaches based on Deep Neural Networks (DNNs) have been proposed. For example, [44] uses an autoencoder-based network architecture, while [40] designs a GAN for watemrark. Those techniques are then generalized to photographs [28] and videos [32]. Notably, there are existing studies focusing on watermarking generative neural networks, such as GANs [8] and image processing networks [25]. Their goal is to safeguard the *intellectual property (IP) of generative models and generated images*, while our method is specifically designed for safeguarding *the copyright of data against potential infringement by these GDMs*. To accomplish their goals, the works [33, 35, 42, 37] embed imperceptible watermarks into every output of a generative model, enabling the defender to determine whether an image was generated by a specific model or not. Various approaches have been employed to inject watermarks, including reformulating the training objectives of the generative models [33], modifying the model's training data [35, 42], or directly conducting watermark embedding to the output images before they are presented to end-users [37].

3. METHOD

In this section, we first formally define the problem and the key notations. Next, we show that the "pattern uniformity" is a key factor for the watermark of generated samples. Based on this, we introduce two essential components of our method, DiffusionShield, i.e., (i) blockwise watermark with pattern uniformity and (ii) joint optimization, and then provide theoretic analysis of pattern uniformity.

3.1 Problem Statement

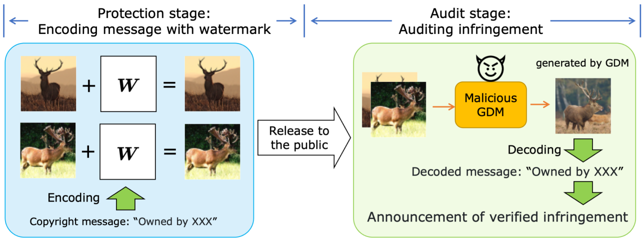

Protection stage:
Encoding message with watermark
Audit stage:
Auditing infringement
W
W
Encoding
Copyright message: "Owned by XXX"
Release to the public
generated by GDM
Malicious GDM
Decoding
Decoded message: "Owned by XXX"
Announcement of verified infringement

Figure 2: An overview of watermarking with two stages.

In this work, we consider two roles: (1) **a data owner** who holds the copyright of the data, releases the data solely for public browsing, and aspires to protect them from being replicated by GDMs, and (2) **a data offender** who employs a GDM on the released data to learn the creative works and infringe the copyright. Besides, since data are often collected from multiple sources to train GDMs in reality, we also consider a scenario where multiple owners protect their copyright against GDMs by encoding their own copyright information into watermarks. We first define the one-owner case, and then extend to the multiple-owner case:

Protection for one-owner case. An image owner aims to release n images, $\{\boldsymbol{X}_{1:n}\}$, strictly for browsing. Each image $\boldsymbol{X}_i$ has a shape of (U, V) where U and V are the height and width, respectively. As shown in Figure 2, the protection process generally comprises two stages: 1) *a protection stage* when the owner encodes the copyright information into the invisible watermark and adds it to the protected data; and 2) *an audit stage* when the owner examines whether a generated sample infringes upon their data. In the following, we introduce crucial definitions and notations.

(1) *The protection stage* happens before the owner releases $\{\boldsymbol{X}_{1:n}\}$ to the public. To protect the copyright, the owner

encodes the copyright message $\boldsymbol{M}$ into each of the invisible watermarks $\{\boldsymbol{W}_{1:n}\}$, and adds $\boldsymbol{W}_i$ into $\boldsymbol{X}_i$ to get a protected data $\tilde{\boldsymbol{X}}_i = \boldsymbol{X}_i + \boldsymbol{W}_i$. $\boldsymbol{M}$ contains information like texts that can signify the owners' unique copyright. The images $\tilde{\boldsymbol{X}}_i$ and $\boldsymbol{X}$ appear similar in human eyes with a small watermark budget $\|\boldsymbol{W}_i\|_p \leq \epsilon$. Instead of releasing $\{\boldsymbol{X}_{1:n}\}$, the owner releases the protected $\{\tilde{\boldsymbol{X}}_{1:n}\}$ for public browsing.

(2) *The audit stage* refers to that the owner finds suspicious images which potentially offend the copyright of their images, and they scrutinize whether these images are generated from their released data. We assume that the data offender collects a dataset $\{\boldsymbol{X}^{\mathcal{G}}_{1:N}\}$ that contains the protected images $\{\tilde{\boldsymbol{X}}_{1:n}\}$, i.e. $\{\tilde{\boldsymbol{X}}_{1:n}\} \subset \{\boldsymbol{X}^{\mathcal{G}}_{1:N}\}$ where N is the total number of both protected and unprotected images ($N > n$). The data offender then trains a GDM, $\mathcal{G}$, from scratch to generate images, $\boldsymbol{X}_{\mathcal{G}}$. If $\boldsymbol{X}_{\mathcal{G}}$ contains the copyright information of the data owner, once $\boldsymbol{X}_{\mathcal{G}}$ is inputted to a decoder $\mathcal{D}$, the copyright message should be decoded by $\mathcal{D}$.

Protection for multi-owner case. When there are K owners to protect their distinct data, we denote their sets of images as $\{\boldsymbol{X}^k_{1:n}\}$ where $k = 1, ..., K$. Following the methodology of one-owner case, each owner can re-use the same encoding process and decoder to encode and decode distinct messages in different watermarks, $\boldsymbol{W}^k_i$, which signifies their specific copyright messages $\boldsymbol{M}^k$. The protected version of images is denoted by $\tilde{\boldsymbol{X}}^k_i = \boldsymbol{X}^k_i + \boldsymbol{W}^k_i$. Then the protected images, $\{\tilde{\boldsymbol{X}}^k_{1:n}\}$, can be released by their respective owners for public browsing, ensuring their copyright is maintained. More details about the two cases are shown in Appendix A.

3.2 Pattern Uniformity

In this subsection, we uncover one important factor "*pattern uniformity*" which could be an important reason for the failure of existing watermark techniques. Previous studies [25, 29, 6] observe that GDMs tend to learn data samples from high probability density regions in the data space and ignore the low probability density regions. However, many existing watermarks such as FRQ [18] and HiDDeN [44] can only generate distinct watermarks for different data samples. Since their generated watermarks are dispersed, these watermarks cannot be effectively extracted and learned.

Observing the above, we formally define the "pattern uniformity" as the consistency of different watermarks injected for different samples:

$$Z = 1 - \frac{1}{n}\sum_{i=1}^{n}\left\| \frac{\boldsymbol{W}_i}{\|\boldsymbol{W}_i\|_2} - \boldsymbol{W}_{mean} \right\|_2, \qquad (1)$$

$$\text{where } \boldsymbol{W}_{mean} = \frac{1}{n}\sum_{i=1}^{n}\frac{\boldsymbol{W}_i}{\|\boldsymbol{W}_i\|_2}.$$

The notation Z corresponds to the standard deviation of normalized watermarks. We further conduct experiments to illustrate the importance of this "*pattern uniformity*". In the experiment shown in Figure 3, we test the ability of DDPM in learning watermarks with different pattern uniformity. The watermarks $\boldsymbol{W}_i$ are random pictures whose pixel value is re-scaled by the budget σ, and the watermarked images are $\tilde{\boldsymbol{X}}_i = \boldsymbol{X}_i + \sigma \times \boldsymbol{W}_i$. More details about the settings for this watermark and the detector can be found in Appendix D.1.

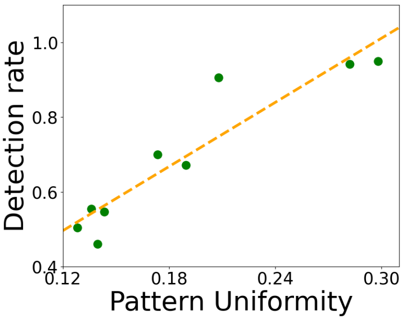


Figure 3: Uniformity vs. watermark detection rate.

Figure 3 illustrates a positive correlation between watermark detection rate in the GDM-generated images and pattern uniformity, which implies that pattern uniformity improves watermark reproduction. Based on pattern uniformity, in Section 3.3 and 3.4, we introduce how to design DiffusionShield, and in Section 3.5, we provide a theoretic analysis of the pattern uniformity based on two examples to justify that the watermarks will be first learned prior to other sparse hidden features and, thus, provide an effective protection.

3.3 Watermarks and Decoding Watermarks

In this subsection, we introduce our proposed approach, referred as DiffusionShield. This model is designed to resolve the problem of inadequate reproduction of prior watermarking approaches in generated images. It adopts a blockwise watermarking approach to augment pattern uniformity, which improves the reproduction of watermarks in generated images and enhances flexibility.

Blockwise watermarks. In DiffusionShield, to strengthen the pattern uniformity in $\{\boldsymbol{W}_{1:n}\}$, we use the same watermark $\boldsymbol{W}$ for each $\boldsymbol{X}_i$ from the same owner. The sequence of *basic patches* encodes the textual copyright message $\boldsymbol{M}$ of the owner. In detail, $\boldsymbol{M}$ is first converted into a sequence of binary numbers by predefined rules such as ASCII. To condense the sequence's length, we convert the binary sequence into a B-nary sequence, denoted as $\{\boldsymbol{b}_{1:m}\}$, where m is the message length and B-nary denotes different numeral systems like quarternary ($B = 4$) and octal ($B = 8$). Accordingly, DiffusionShield partitions the whole watermark $\boldsymbol{W}$ into a sequence of m patches, $\{\boldsymbol{w}_{1:m}\}$, to represent $\{\boldsymbol{b}_{1:m}\}$. Each patch is chosen from a candidate set of basic patch $\{\boldsymbol{w}^{(1:B)}\}$. The set $\{\boldsymbol{w}^{(1:B)}\}$ has B basic patch candidates with a shape (u, v), which represent different values of the B-nary bits. The sequence of $\{\boldsymbol{w}_{1:m}\}$ denotes the B-nary bits $\{\boldsymbol{b}_{1:m}\}$ derived from $\boldsymbol{M}$.

For example, in Figure 4, we have 4 patches ($B = 4$), and each of the patches has a unique pattern which represents 0, 1, 2, and 3. To encode the copyright message $\boldsymbol{M}$ = "Owned by XXX" (as an example, where $\boldsymbol{M}$ can be any arbitrary message), we first convert it into binary sequence "01001111 01110111..." based on ASCII, and transfer it into quarternary sequence $\{\boldsymbol{b}_{1:m}\}$, "103313131232...". (The sequence length m should be less or equal to 8×8, since there are only 8×8 patches in Figure 4.) Then we concatenate these basic patches in the order of $\{\boldsymbol{b}_{1:m}\}$ for the complete watermark $\boldsymbol{W}$ and add $\boldsymbol{W}$ to each image from the data owner. Once the offender uses GDMs to learn from it, the watermarks will appear in generated images, serving as an

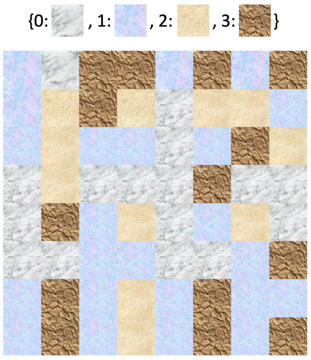


Figure 4: An 8×8 sequence of basic patches encoded with message "103313131...". Different patterns represent different basic patches.

evidence of infringement.

Decoding the watermarks. DiffusionShield employs a decoder $\mathcal{D}_\theta$ by classification in patches, where θ is the parameters. $\mathcal{D}_\theta$ can classify $\boldsymbol{w}_i$ into a bit $\boldsymbol{b}_i$. The decoder $\mathcal{D}_\theta$ accepts a watermarked image block, $\boldsymbol{x}_i + \boldsymbol{w}_i$, as input and outputs the bit value of $\boldsymbol{w}_i$, i.e., $\boldsymbol{b}_i = \mathcal{D}_\theta(\boldsymbol{x}_i + \boldsymbol{w}_i)$. The suspect generated image is partitioned into a sequence $\{(\boldsymbol{x} + \boldsymbol{w})_{1:m}\}$, and then is classified into $\{\boldsymbol{b}_{1:m}\} = \{\mathcal{D}_\theta(\boldsymbol{x}_i + \boldsymbol{w}_i)|i = 1, ..., m\}$ in a patch-by-patch manner. If $\{\boldsymbol{b}_{1:m}\}$ is the B-nary message that we embed into the watermark, we can accurately identify the owner of the data, and reveal the infringement.

Remarks. Since we assign the same watermark $\boldsymbol{W}$ to each image of one user, the designed watermark evidently has higher uniformity. Besides, DiffusionShield shows remarkable flexibility when applied to multiple-owner scenarios as basic patches and decoder can be reused by new owners.

3.4 Jointly Optimize Watermark and Decoder

While pattern uniformity facilitates the reproduction of watermarks in GDM-generated images, it does not guarantee the detection performance of the decoder $\mathcal{D}_\theta$. Therefore, we further propose a joint optimization method to search for the optimal basic patch patterns and obtain the optimized detection decoder simultaneously. Ideally, the basic patches and the decoder should satisfy:

$$\boldsymbol{b}^{(i)} = \mathcal{D}_\theta(\boldsymbol{p} + \boldsymbol{w}^{(i)}) \text{ for } \forall\ i \in \{1, 2, ..., B\}, \tag{2}$$

where $\boldsymbol{w}^{(i)}$ is one of the B basic patch candidates, $\boldsymbol{b}^{(i)}$ is the correct label for $\boldsymbol{w}^{(i)}$, and $\boldsymbol{p}$ can be a random block with the same shape as $\boldsymbol{w}^{(i)}$ cropped from any image. The ideal decoder, capable of accurately predicting all the watermarked blocks, ensures that all embedded information can be decoded from the watermark. To increase the detection performance, we simultaneously optimize the basic patches and the decoder using the following bi-level objective:

$$\min_{\boldsymbol{w}^{1:B}} \min_{\theta} \mathbb{E}\left[\sum_{i=1}^{B} L_{\text{CE}}\left(\mathcal{D}_\theta\left(\boldsymbol{p} + \boldsymbol{w}^{(i)}\right), \boldsymbol{b}^{(i)}\right)\right] \text{ s.t. } \|\boldsymbol{w}^{(i)}\|_\infty \leq \epsilon,$$

where L_{CE} is the cross-entropy loss for the classification. The l_∞ budget is constrained by ϵ. To reduce the number of categories of basic patches, we set $\boldsymbol{w}^{(1)} = \mathbf{0}$, which means that the blocks without watermark should be classified as $\boldsymbol{b} = 1$. Thus, the bi-level optimization can be rewritten as:

$$\begin{cases} \theta^* = \underset{\theta}{\arg\min}\ \mathbb{E}\left[\sum_{i=1}^{B} L_{\text{CE}}\left(\mathcal{D}_\theta\left(\boldsymbol{p} + \boldsymbol{w}^{(i)}\right), \boldsymbol{b}^{(i)}\right)\right] \\ \boldsymbol{w}^{(2:B),*} = \underset{\boldsymbol{w}^{(2:B)}}{\arg\min}\ \mathbb{E}\left[\sum_{i=2}^{B} L_{\text{CE}}\left(\mathcal{D}_{\theta^*}\left(\boldsymbol{p} + \boldsymbol{w}^{(i)}\right), \boldsymbol{b}^{(i)}\right)\right] \\ \qquad \text{s.t.} \|\boldsymbol{w}^{(i)}\|_\infty \leq \epsilon. \end{cases} \tag{3}$$

The upper-level objective aims to increase the performance of $\mathcal{D}_\theta$, while the lower-level objective optimizes the basic patches to facilitate their detection by the decoder. By the two levels of objectives, the basic patches and decoder potentially promote each other to achieve higher accuracy on a smaller budget. To ensure basic patches can be adapted to various image blocks and increase their flexibility, we use randomly cropped image blocks as the host images in the training process of basic patches and decoders. More details about the algorithm can be found in Appendix B.

3.5 Theoretic Analysis of Pattern Uniformity

In this subsection, we provide theoretic analysis with two examples, a linear regression model for supervised task, and a multilayer perceptron (MLP) with a general loss function (which can be a **generation** task), to justify that watermarks with pattern uniformity are stronger than other features, and machine learning models can learn features from watermarks earlier and more easily regardless of the type of tasks. Following the same idea, DiffusionShield provides an effective protection since GDMs have to learn watermarks first if they want to learn from protected images.

For both examples, we use the same assumption for the features in the watermarked dataset. For simplicity, we assume the identical watermark is added onto each sample in the dataset. We impose the following data assumption, which is extended from the existing sparse coding model [20, 17, 2].

Assumption 3.1 (Sparse coding model with watermark)**.** The observed data is $\boldsymbol{Z} = \boldsymbol{M}\boldsymbol{S}$, where $\boldsymbol{M} \in \mathbb{R}^{d \times d}$ is a unitary matrix, and $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_d)^\top \in \mathbb{R}^d$ is the hidden feature composed of d sparse features:

$$P(\boldsymbol{s}_i \neq 0) = p, \text{and } \boldsymbol{s}_i^2 = \mathcal{O}(1/pd) \text{ when } \boldsymbol{s}_i \neq 0. \tag{4}$$

$\|\cdot\|$ is L_2 norm. For $\forall i \in [d]$, $\mathbb{E}[\boldsymbol{s}_i] = 0$. The watermarked data is $\tilde{\boldsymbol{Z}} = \boldsymbol{M}\boldsymbol{S} + \boldsymbol{\delta}$, and $\boldsymbol{\delta}$ is a constant watermark vector for all the data samples because of pattern uniformity.

For the linear regression task, $\boldsymbol{Y} = \boldsymbol{S}^\top \boldsymbol{\beta} + \boldsymbol{\epsilon}$ is the ground truth label, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2)$ is the noise and $\beta_i = \Theta(1)$ so that $Y^2 = \mathcal{O}_p(1)$. We represent the linear regression model as $\hat{\boldsymbol{Y}} = \tilde{\boldsymbol{Z}}^\top \boldsymbol{w}$, using the watermark data $\tilde{\boldsymbol{Z}}$, where $\boldsymbol{w} \in \mathbb{R}^{1 \times d}$ is the parameter to learn. The mean square error (MSE) loss for linear regression task can be represented as

$$L(\mathbf{w}) = (\tilde{\boldsymbol{Z}}^\top \mathbf{w} - \boldsymbol{S}^\top \boldsymbol{\beta} - \boldsymbol{\epsilon})^2.$$

Given the above problem setup, we have following result: Consider the initial stage of the training, i.e., $\mathbf{w}$ is initialized with $\mathrm{w}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$. With Assumption 3.1, the gradient, with respect to $\mathbf{w}$, of MSE loss for the linear regression model given infinite samples can be derived as

$$\mathbb{E}\left[\frac{\partial L}{\partial \mathbf{w}}\right] = \mathbb{E}\left[A(\boldsymbol{S})\right] + \mathbb{E}\left[B(\boldsymbol{\delta})\right], \tag{5}$$

where $\mathbb{E}[A(\boldsymbol{S})]$ is the hidden term that contains the gradient terms from hidden features, and $\mathbb{E}[B(\boldsymbol{\delta})]$ is the watermark term that contains the gradient terms from the watermark. There are three observations. First, watermark is learned prior to other hidden features after initialization. If $\|\boldsymbol{\delta}\| \gg 1/\sqrt{d}$, then with high probability w.r.t. the initialization, $\mathbb{E}\|B(\boldsymbol{\delta})\| \gg \mathbb{E}\|A(\boldsymbol{S})\|$, and $\mathbb{E}\|B(\boldsymbol{\delta})\|$ is maximized with the best uniformity. Second, since $\|\boldsymbol{\delta}\| \ll 1/\sqrt{pd}$, the watermark $\boldsymbol{\delta}$ will be much smaller than any active hidden feature. Finally, when the training converges, the final trained model does not forget $\boldsymbol{\delta}$. (The proof is in Appendix C.1.)

In addition to the linear regression task, we extend our analysis to neural networks with a general loss to further explain the feasibility of the intuition for a generative task. We follow Assumption 3.1 and give the toy example for neural networks: We use an MLP with $\tilde{\boldsymbol{Z}}$ as input to fit a general loss $L(\boldsymbol{\mathcal{W}}, \tilde{\boldsymbol{Z}})$. The loss $L(\boldsymbol{\mathcal{W}}, \tilde{\boldsymbol{Z}})$ can be a classification or generation task. The notation $\boldsymbol{\mathcal{W}}$ is the parameter of it, and $\boldsymbol{\mathcal{W}}_1$ is the first layer of $\boldsymbol{\mathcal{W}}$. Under mild assumptions, we can derive the gradient with respect to each neuron in $\boldsymbol{\mathcal{W}}_1$ into hidden feature term and watermark term as Eq. 5. When $1/\sqrt{d} \ll \|\boldsymbol{\delta}\| \ll 1/\sqrt{pd}$, the watermark term will have more influence and be learned prior to other hidden features in the first layer even though the watermark has a much smaller norm than each active hidden feature. (The proof can be found in Appendix C.2.)

With the theoretical analysis in the above two examples, we justify that the watermark with high pattern uniformity is easier/earlier to be learned than other sparse hidden features. It suggests if the authorized people use GDM to learn from the protected images, the GDM will first learn the watermarks before the data distribution. Therefore, our method can provide an effective protection agaist GDM. We also provide empirical evidence to support this analysis in Appendix C.3.

4. EXPERIMENT

In this section, we assess the efficacy of DiffusionShield across various budgets, datasets, and protection scenarios. We first introduce our experimental setups in Section 4.1. In Section 4.2, we evaluate the watermark's performance in terms of its accuracy and invisibility. Then we investigate the watermark's flexibility and efficacy in multiple-user cases, the impact of budget and watermark rate, the watermark's generalization to fine-tuning GDMs, capacity for message length and robustness, from Section 4.3 to 4.7. We also evaluate the quality of generated images and in Appendix F.4.

4.1 Experimental Settings

Datasets, baselines and GDM. We conduct the experiments using four datasets and compare DiffusionShield with four baseline methods. The datasets include CIFAR10 and CIFAR100, both with $(U, V) = (32, 32)$, STL10 with $(U, V) = (64, 64)$ and ImageNet-20 with $(U, V) = (256, 256)$. The baseline methods include Image Blending (IB) which is a simplified version of DiffusionShield without joint optimization, DWT-DCT-SVD based watermarking in the frequency domain (FRQ) [18], HiDDeN [44], and DeepFake Fingerprint Detection (DFD) [35] (which is designed for DeepFake Detection and adapted to our data protection goal). In the audit stage, we use the improved DDPM [19] as the GDM to train on watermarked data. More details about the baselines are shown in Appendix D.4.

Evaluation metrics. In our experiments, we generate T images from each GDM and decode copyright messages from them. We compare the effectiveness of watermarks in terms of their invisibility, the decoding performance, and the capacity to embed longer messages:

- **(Perturbation) Budget.** We use the LPIPS [39] metric together with l_2 and l_∞ differences to measure the visual discrepancies between the original and watermarked images. The lower values of these metrics indicate better invisibility.

- **(Detection) Accuracy.** Following [35] and [43], we apply bit accuracy to evaluate the correctness of detected messages encoded. To compute bit accuracy, we transform the ground truth B-nary message $\{\boldsymbol{b}_{1:m}\}$ and the decoded $\{\hat{\boldsymbol{b}}_{1:m}\}$ back into binary messages $\{\boldsymbol{b}'_{1:m \log_2 B}\}$ and $\{\hat{\boldsymbol{b}}'_{1:m \log_2 B}\}$. The bit accuracy for one watermark is

$$\text{Bit-Acc} \equiv \frac{1}{m \log_2 B} \sum_{k=1}^{m \log_2 B} \mathbb{1}\left(\boldsymbol{b}'_{1:m \log_2 B} = \hat{\boldsymbol{b}}'_{1:m \log_2 B}\right).$$

  The worst bit accuracy is expected to be 50%, which is equivalent to random guessing.

- **Message length.** The length of the encoded message reflects the capacity of encoding. To ensure the accuracy of FRQ and HiDDeN, we use a 16-bit and a 32-bit message for CIFAR images and a 64-bit one for STL10. For others, we encode 128 bits into CIFAR, 512 bits into STL10 and 256 bits into ImageNet.

Implementation details. We set $(u, v) = (4, 4)$ as the shape of the basic patches and set $B = 4$ for quarternary messages. We use ResNet [10] as the decoder to classify different basic patches. For the joint optimization, we use 5-step PGD [16] with $l_\infty \leq \epsilon$ to update the basic patches and use SGD to optimize the decoder. As mentioned in Section 3.1, the data offender may collect and train the watermarked images and non-watermarked images together to train GDMs. Hence, in all the datasets, we designate one random class of images as watermarked images, while treating other classes as unprotected images. To generate images of the protected class, we either 1) use a **class-conditional** GDM to generate images from the specified class, or 2) apply a classifier to filter images of the protected class from the **unconditional** GDM's output. The bit accuracy on unconditionally generated images may be lower than that of the conditional generated images since object classifiers cannot achieve 100% accuracy. In the joint optimization, we use SGD with 0.01 learning rate and 5×10^{-4} weight decay to train the decoder and we use 5-step PGD with step size to be 1/10 of the L_∞ budget to train the basic patches. More details are presented in Appendix D.3.

4.2 Results on Protection Performance

In this subsection, we show that DiffusionShield provides protection with high bit accuracy and good invisibility in Table 1. We compare on two groups of images: (1) the originally released images with watermarks (**Released**) and (2) the generated images from class-conditional GDM or unconditional GDM trained on watermarked data (**Cond.** and **Uncond.**). Based on Table 1, we can see:

Table 1: Bit accuracy (%) and budget of the watermark

			IB	FRQ	HiDDeN	DFD	DiffusionShield (ours)			
CIFAR10	Budget	l_∞	7/255	13/255	65/255	28/255	**1/255**	2/255	4/255	8/255
		l_2	0.52	0.70	2.65	1.21	**0.18**	0.36	0.72	1.43
		LPIPS	0.01582	0.01790	0.14924	0.07095	**0.00005**	0.00020	0.00120	0.01470
	Accuracy	Released	87.2767	99.7875	99.0734	95.7763	99.6955	99.9466	99.9909	**99.9933**
		Cond.	87.4840	57.7469	98.9250	93.5703	99.8992	99.9945	**100.0000**	99.9996
		Uncond.	81.4839	55.6907	**97.1536**	89.1977	93.8186	95.0618	96.8904	96.0877
	Pattern Uniformity		0.963	0.056	0.260	0.236	0.974	0.971	0.964	0.954
CIFAR100	Budget	l_∞	7/255	14/255	75/255	44/255	**1/255**	2/255	4/255	8/255
		l_2	0.52	0.69	3.80	1.58	**0.18**	0.36	0.72	1.43
		LPIPS	0.00840	0.00641	0.16677	0.03563	**0.00009**	0.00013	0.00134	0.00672
	Accuracy	Released	84.6156	99.5250	99.7000	96.1297	99.5547	99.9297	99.9797	**99.9922**
		Cond.	54.3406	54.4438	95.8640	90.5828	52.0078	64.3563	99.8000	**99.9984**
		Uncond.	52.6963	54.6370	81.9852	79.0234	52.9576	53.1436	85.7057	**91.2946**
	Pattern Uniformity		0.822	0.107	0.161	0.180	0.854	0.855	0.836	0.816
STL10	Budget	l_∞	8/255	14/255	119/255	36/255	**1/255**	2/255	4/255	8/255
		l_2	1.09	1.40	7.28	2.16	**0.38**	0.76	1.51	3.00
		LPIPS	0.06947	0.02341	0.32995	0.09174	**0.00026**	0.00137	0.00817	0.03428
	Accuracy	Released	92.5895	99.5750	97.2769	94.2813	99.4969	99.9449	99.9762	**99.9926**
		Cond.	96.0541	54.3945	96.5164	94.7236	95.4848	99.8164	99.8883	**99.9828**
		Uncond.	89.2259	56.3038	91.3919	91.8919	82.5841	93.4693	**96.1360**	95.0586
	Pattern Uniformity		0.895	0.071	0.155	0.203	0.924	0.921	0.915	0.907
ImageNet-20	Budget	l_∞	-	20/255	139/255	88/255	**1/255**	2/255	4/255	8/255
		l_2	-	5.60	25.65	21.68	**1.17**	2.33	4.64	9.12
		LPIPS	-	0.08480	0.44775	0.30339	**0.00019**	0.00125	0.00661	0.17555
	Accuracy	Released	-	99.8960	98.0625	99.3554	99.9375	99.9970	99.9993	**100.0000**
		Cond.	-	50.6090	98.2500	81.3232	53.6865	53.7597	99.9524	**100.0000**
	Pattern Uniformity		-	0.061	0.033	0.041	0.941	0.930	0.908	0.885

First, DiffusionShield can protect the images with the highest bit accuracy and the lowest budget among all the methods. For example, on CIFAR10 and STL10, with all the budgets from 1/255 to 8/255, DiffusionShield achieves almost 100% bit accuracy on released images and conditionally generated images, which is better than all the baseline methods. Even constrained by the smallest budget with an l_∞ norm of 1/255, DiffusionShield still achieves a high successful reproduction rate. On CIFAR100 and ImageNet, DiffusionShield with an l_∞ budget of 4/255 achieves a higher bit accuracy in generated images with a much lower l_∞ difference and LPIPS than baseline methods. For baselines, FRQ cannot be reproduced by GDM, while HiDDeN and DFD require a much larger perturbation budget over DiffusionShield (Image examples are shown in Appendix E). The accuracy of IB is much worse than the DiffusionShield with 1/255 budget on CIFAR10 and STL10. To explain IB, without joint optimization, the decoder cannot perform well on released images and thus cannot guarantee its accuracy on generated images, indicating the importance of joint optimization. To further illustrate the invisibility of DiffusionShield, we demonstrate a visualization of its impact on the image feature space in Appendix G. It clearly shows that our method introduces negligible alterations to images' features.

Second, enforcing pattern uniformity can promote the reproduction of watermarks in generated images. In Table 1, we can see that the bit accuracy of the conditionally generated images watermarked by DiffusionShield is as high as that of released images with a proper budget. In addition to DiffusionShield, IB's accuracy in released data and conditionally generated data are also similar. This is because IB is a simplified version of our method without joint optimization and also has high pattern uniformity. In contrast, other methods without pattern uniformity all suffer from a drop of accuracy from released images to conditionally generated images, especially FRQ, which has pattern uniformity lower than 0.11 and an accuracy level on par with a random guess. This implies that the decoded information in watermarks with high pattern uniformity (e.g., IB and ours in CIFAR10 are higher than 0.95) does not change much from released images to generated images and the watermarks can be exactly and easily captured by GDM. Notably, the performance drop on CIFAR100 and ImageNet in 1/255 and 2/255 is also partially due to the low watermark rate. In fact, both a small budget and a low watermark rate can hurt the reproduction of watermarks in generated images. We provide further analysis on the influence of budget and watermark rate in Section 4.3

In addition to the four baselines, we have also compared DiffusionShield with three other watermarking approaches, i.e., IGA [36], MBRS [13], and CIN [15]. Overall, DiffusionShield still demonstrates a better trade-off between bit accuracy and budget compared to the other methods. The comparison results can be found in Appendix F.1.

4.3 Flexibility and Efficacy in Multiple-user Case

In this subsection, we demonstrate that DiffusionShield is flexible to be transferred to new users while maintaining good protection against GDMs. We assume that multiple copy-

Table 2: Average bit accuracy (%) across different numbers of copyright owners (on class-conditional GDM).

owners	CIFAR-10	CIFAR-100
1	100.0000	99.8000
4	99.9986	99.9898
10	99.9993	99.9986

right owners are using DiffusionShield to protect their images, and different copyright messages should be encoded into the images from different copyright owners. In Table 2, we use one class in the dataset as the first owner and the other classes as the new owners. The basic patches (with 4/255 l_∞ budget) and decoder are optimized on the first class and re-used to protect the new classes. Images within the same class have the same message embedded, while images from different classes have distinct messages embedded in them. After reordering the basic patches for different messages, transferring from one class to the other classes does not take any additional calculation, and is efficient. We train class-conditional GDM on all of the protected data and get the average bit accuracy across classes. As shown in Table 2, on both CIFAR10 and CIFAR100, when we reorder the basic patches to protect the other 3 classes or 9 classes, the protection performance is almost the same as the one class case, with bit accuracy all close to 100%. Besides flexibility, our watermarks can protect each of the multiple users and can distinguish them clearly even when their data are mixed by the data offender.

4.4 Impact of Budget and Watermark Rate

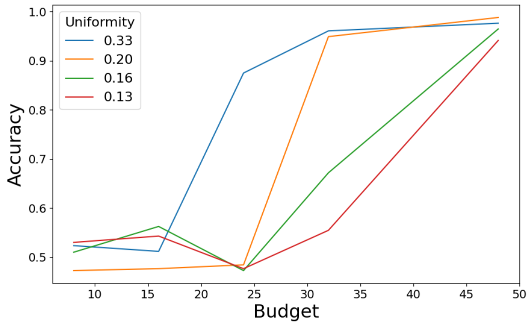


Figure 5: The change of bit accuracy under different budgets

As mentioned in Section 4.2, the watermark reproduction in generated images is highly influenced by the budget and watermark rate. In this subsection, we provide more analysis on the impact of this two aspects.

Impact of Budget. In Figure 5, we follow the same setting in Section 3.2 and show the change of bit accuracy when adopting different budgets and using watermark with different levels of pattern uniformity. From the figure, we can see that with the same uniformity, the watermark detection accuracy increase as the budget increases, indicating that a larger budget can enhance the watermark's reproduction on generated images. This can also be validated by the results in Table 4.2 that the bit accuracy of budget 1/255 and 2/255 on CIFAR100 is lower than that of 4/255 and 8/255. Meanwhile, the results in Figure 5 also indicates that with a higher pattern uniformity, the bit accuracy of the watermark detection is also higher, which is consistent with the analysis in Section 3.2.

Impact of Watermark Rate. In Figure 6, we show the bit accuracy of DiffusionShield while controlling the proportion of the watermarked images in the training set of GDM. From the figure, we can see that the bit accuracy rises from around 53% to almost 100% when the watermark rate increases from 0.05% to 10%, indicating that the degree of watermark reproduction is greatly affected by the watermark rate. Nevertheless, even with a watermark rate as low as 5%, DiffusionShield can achieve effective protection with a bit accuracy higher than 90%.

In addition to the single-owner scenario, in Figure 7, we check the performance of DiffusionShield across different numbers of users, given a small watermark rate and a low budget for each user. Notably, although each user has a distinct watermark message, they use the same set of basic patches to form the watermark. This potentially enhances the reproducibility of watermark by ensuring a high frequency of identical watermark patches throughout the entire GDM training set. In the experiments of Figure 7, we consider that there are K owners and the images of each owner compose 1% of the collected training data. From the figure we can see that, as the number of owners increases from 1 to 20, the average accuracy increases from around 64% to nearly 100%. This observation suggests that, despite the challenges posed by low watermark rates and limited budgets, applying DiffusionShield in a multi-user scenario results in strong performance.

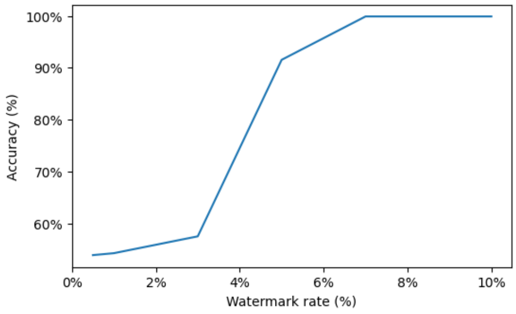


Figure 6: The change of bit accuracy with different watermark rates (budget=1/255)

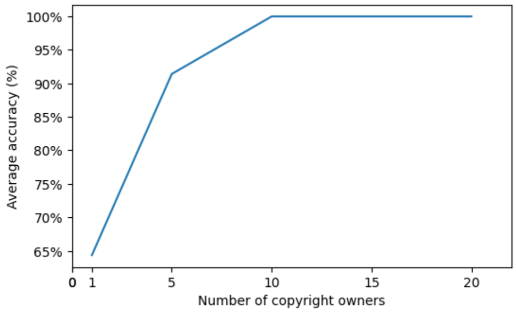


Figure 7: The change of bit accuracy with different numbers of copyright owners (budget=2/255)

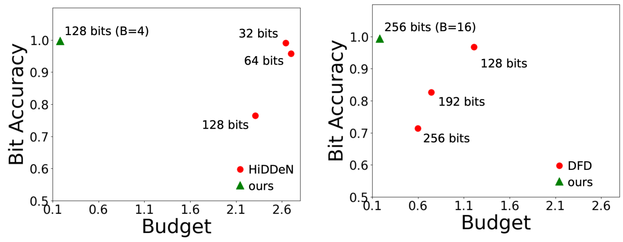
4.5 Generalization to Fine-tuning GDMs

In this subsection, we test the performance of our method when generalized to the fine-tuning GDMs [24], which is also a common strategy for learning and generating images. Fine-tuning is a more difficult task compared to the training-from-scratch setting because fine-tuning only changes the GDM parameters to a limited extent. This change may be not sufficient to learn all the features in the fine-tuned dataset, therefore, the priority by pattern uniformity becomes even more important. To better generalize our method to the fine-tuning case, we enhance the uniformity in hidden space instead of pixel space, and limit l_2 norm instead of l_∞ norm. More details of fine-tuning and its experiment settings can be found in Appendix D.6. We assume that the data offender fine-tunes Stable Diffusion [24] to learn the style of *pokemon-blip-captions* dataset [21]. In Table 3, we compare the budget and bit accuracy of our method with three baselines. The observation is similar to that in Table 1. Although FRQ has a lower budget than ours, the bit accuracy on generated images is much worse. DFD has bit accuracy of 90.31%, but the budget is three times of ours. HiDDeN is worse than ours in both budget and bit accuracy. We further investigate the impact of the watermark on the hidden space in Appendix G, which aligns with the metrics presented in Table 3. In summary, our method has the highest accuracy in both released and generated data.

Table 3: Bit Acc. (%) in fine-tuning.

	FRQ	DFD	HiDDeN	Ours
l_2	8.95	61.30	63.40	21.22
Released	88.86	99.20	89.48	99.50
Generated	57.13	90.31	60.16	92.88

4.6 Capacity for Message Length



(a) HiDDeN & ours (1/255)

(b) DFD & ours (1/255)

Figure 8: Bit acc. and l_2 of different message lengths

The capacity of embedding longer messages is important for watermarking methods since encoding more information can provide more conclusive evidence of infringement. In this subsection, we show the superiority of DiffusionShield over other methods in achieving high watermark capacity. Figure 8 shows the bit accuracy and l_2 budgets of watermarks with different message lengths on the released protected images in CIFAR10. In Figure 8(a), we can see that HiDDeN consistently requires a large budget across varying message lengths, and its accuracy diminishes to 77% at 128 bits. Conversely, DiffusionShield maintains nearly 100% accuracy at 128 bits, even with a much smaller budget. Similarly, in Figure 8(b), ours maintains longer capacity with better accuracy and budget than DFD. This indicate that DiffusionShield has much greater capacity than HiDDeN and DFD and can maintain good performance even with increased message lengths.

Table 4: Bit Acc. (%) under corruptions

	DFD	HiDDeN	Ours
No corrupt	93.57	98.93	99.99
Gaussian noise	68.63	83.59	81.93
Low-pass filter	88.94	81.05	99.86
Greyscale	50.82	97.81	99.81
JPEG comp.	62.52	74.84	94.45
Resize (Larger)	93.20	79.69	99.99
Resize (Smaller)	92.38	83.13	99.30
Wm. removal	91.11	82.20	99.95

4.7 Robustness of DiffusionShield

Robustness of watermarks is important since there is a risk that the watermarks may be distorted by disturbances, such as image corruption due to deliberate post-processing activities during the images' circulation, the application of speeding-up sampling methods in the GDM [27], or different training hyper-parameters used to train GDM. This subsection demonstrate that DiffusionShield is robust in accuracy on generated images when corrupted. In Appendix F.2 and F.3, we show similar conclusions when sampling procedure is fastened and hyper-parameters are changed.

We consider Gaussian noise, low-pass filter, greyscale, JPEG compression, resizing, and the watermark removal attack proposed by [41] to test the robustness of DiffusionShield against image corruptions. Details about the severity of the corruptions are shown in Appendix D.5. Different from the previous experiments, during the protection stage, we augment our method by incorporating corruptions into the joint optimization. Each corruption is employed after the basic patches are added to the images. Table 4 shows the bit accuracy of DiffusionShield (with 8/255 l_∞ budget) on corrupted generated images. It maintains accuracy above 99.8% under Greyscale, low-pass filter, resizing to larger size and watermark removal attack, nearly matching the accuracy achieved without corruption. In other corruptions, our method performs better than baselines except HiDDeN in Gaussian noise. In contrast, DFD has a significant reduce in Gaussian noise, Greyscale and JPEG compression, and HiDDeN shows a poor performance under low-pass filter and JPEG Compression. From these results, we can see that DiffusionShield is robust against image corruptions.

5. CONCLUSION

In this paper, we introduce DiffusionShield, a watermark to protect data copyright, which is motivated by our observation that the pattern uniformity can effectively assist the watermark to be captured by GDMs. By enhancing pattern uniformity and leveraging a joint optimization method, DiffusionShield successfully secures copyright with better accuracy and a smaller budget. Theoretic analysis and empirical results demonstrate the superior performance of DiffusionShield.

References

[1] Zeyuan Allen-Zhu and Yuanzhi Li. Feature purification: How adversarial training performs robust deep learning. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 977–988. IEEE, 2022.

[2] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

[3] Jimmy Ba, Murat Erdogdu, Taiji Suzuki, Denny Wu, and Tianzong Zhang. Generalization of two-layer neural networks: An asymptotic viewpoint. In *International conference on learning representations*, 2019.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[5] Ingemar Cox, Matthew Miller, Jeffrey Bloom, and Chris Honsinger. Digital watermarking. *Journal of Electronic Imaging*, 11(3):414–414, 2002.

[6] Giannis Daras, Yuval Dagan, Alexandros G Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. *arXiv preprint arXiv:2302.09057*, 2023.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[9] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, pages 87–102. Springer, 2016.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[13] Zhaoyang Jia, Han Fang, and Weiming Zhang. Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression. In *Proceedings of the 29th ACM international conference on multimedia*, pages 41–49, 2021.

[14] Ashwani Kumar. A review on implementation of digital image watermarking techniques using lsb and dwt. *Information and Communication Technology for Sustainable Development: Proceedings of ICT4SD 2018*, pages 595–602, 2020.

[15] Rui Ma, Mengxi Guo, Yi Hou, Fan Yang, Yuan Li, Huizhu Jia, and Xiaodong Xie. Towards blind watermarking: Combining invertible and non-invertible mechanisms. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1532–1542, 2022.

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[17] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(1), 2010.

[18] KA Navas, Mathews Cheriyan Ajay, M Lekshmi, Tampy S Archana, and M Sasikumar. Dwt-dct-svd based watermarking. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, pages 271–274. IEEE, 2008.

[19] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[20] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[21] Justin N. M. Pinkney. Pokemon blip captions. `https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions/`, 2022.

[22] Christine I Podilchuk and Edward J Delp. Digital watermarking: algorithms and applications. *IEEE signal processing Magazine*, 18(4):33–46, 2001.

[23] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[25] Vikash Sehwag, Caner Hazirbas, Albert Gordo, Firat Ozgenel, and Cristian Canton. Generating high fidelity data from low-density regions using diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11492–11501, 2022.

[26] Frank Y Shih and Scott YT Wu. Combinational image watermarking in the spatial and frequency domains. *Pattern Recognition*, 36(4):969–975, 2003.

[27] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[28] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2117–2126, 2020.

[29] Soobin Um and Jong Chul Ye. Don't play favorites: Minority guidance for diffusion models. *arXiv preprint arXiv:2301.12334*, 2023.

[30] James Vincent. Ai art copyright lawsuit: Getty images and stable diffusion. `https://www.theverge.com/2023/2/6/23587393`, Feb 2023. Accessed: May 12, 2023.

[31] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

[32] Xinyu Weng, Yongzhi Li, Lu Chi, and Yadong Mu. High-capacity convolutional video steganography with temporal residual modeling. In *Proceedings of the 2019 on international conference on multimedia retrieval*, pages 87–95, 2019.

[33] Hanzhou Wu, Gen Liu, Yuwei Yao, and Xinpeng Zhang. Watermarking neural networks with watermarked images. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(7):2591–2601, 2020.

[34] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[35] Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 14448–14457, 2021.

[36] Honglei Zhang, Hu Wang, Yuanzhouhan Cao, Chunhua Shen, and Yidong Li. Robust data hiding using inverse gradient attention. *arXiv preprint arXiv:2011.10850*, 2020.

[37] Jie Zhang, Dongdong Chen, Jing Liao, Han Fang, Weiming Zhang, Wenbo Zhou, Hao Cui, and Nenghai Yu. Model watermarking for image processing networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12805–12812, 2020.

[38] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.

[39] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[40] Ru Zhang, Shiqi Dong, and Jianyi Liu. Invisible steganography via generative adversarial networks. *Multimedia tools and applications*, 78:8559–8575, 2019.

[41] Xuandong Zhao, Kexun Zhang, Yu-Xiang Wang, and Lei Li. Generative autoencoders as watermark attackers: Analyses of vulnerabilities and threats. *arXiv preprint arXiv:2306.01953*, 2023.

[42] Yuan Zhao, Bo Liu, Ming Ding, Baoping Liu, Tianqing Zhu, and Xin Yu. Proactive deepfake defence via identity watermarking. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4602–4611, 2023.

[43] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. A recipe for watermarking diffusion models. *arXiv preprint arXiv:2303.10137*, 2023.

[44] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 657–672, 2018.

APPENDIX

A. WATERMARKING PROTECTION FOR MULTIPLE COPYRIGHT OWNERS

As shown in Algorithm 1, to extend the protection from the one-owner case to the multiple-owner case, we first build the watermark protection for one owner and get the corresponding watermark decoder $\mathcal{D}_\theta$ (line 1). Then we use the same procedure (that can be decoded by $\mathcal{D}_\theta$) to watermark all the images from other owners (lines 2 to 4).

Algorithm 1 Watermark protection for multiple copyright owners

```
Input: The number of distinct sets of images to protect, $K$.
    Distinct sets, $\{\boldsymbol{X}^k_{1:n}\}$ and different copyright messages
    for different owners $\boldsymbol{M}^k$, where $k = 1, 2, 3, ..., K$.
Output: Watermarked images $\{\tilde{\boldsymbol{X}}^k_{1:n}\}$, where $k = 1, 2, 3, ..., K$ and the watermark decoder $\mathcal{D}_\theta$.
1: $\{\tilde{\boldsymbol{X}}^1_{1:n}\}, \mathcal{D}_\theta \leftarrow OneOwnerCaseProtection(\{\boldsymbol{X}^1_{1:n}\}, \boldsymbol{M}^1)$
2: for $k = 2$ to K do
3:     $\{\tilde{\boldsymbol{X}}^k_{1:n}\} \leftarrow ReuseEncodingProcess(\{\boldsymbol{X}^k_{1:n}\}, \boldsymbol{M}^k)$
4: end for
5: return $\{\tilde{\boldsymbol{X}}^k_{1:n}\}$, $k = 1, 2, 3, ..., K$ and $\mathcal{D}_\theta$.
```

B. ALGORITHM

As shown in Algorithm 2, the joint optimization is numerically solved by alternately training on the two levels. Every batch is first watermarked and trained on the classifier for upper level objective by gradient descent (line 4 to 6), and then optimized on basic patches for lower level objective by 5-step PGD (line 7 to 9). With the joint optimized basic patches and classifier, we can obtain a robust watermark that can encode different ownership information with a small change on the protected data. This watermark can be easily captured by the diffusion model and is effective for tracking data usage and copyright protection. The clean images $\{\boldsymbol{X}_{1:n}\}$ for input of the algorithm is not necessary to be the images that we want to protect. The random cropped image blocks can help the basic patches to fit different image blocks and then increase the flexibility.

Algorithm 2 Joint optimization on $\{\boldsymbol{w}^{(1:B)}\}$ and $\mathcal{D}_\theta$

```
Input: Initialized basic patches $\{\boldsymbol{w}^{(1:B)}_{(0)}\}$, clean images
    $\{\boldsymbol{X}_{1:n}\}$, upper and lower level objectives in Eq. 3,
    $\mathcal{L}_{\text{upper}}$, $\mathcal{L}_{\text{lower}}$, watermark budget $\epsilon$, decoder learning
    rate $r$, batch size $bs$, PGD step $\alpha$ and epoch $E$.
Output: Optimal $\{\boldsymbol{w}^{(1:B),*}\}$ and $\theta^*$.
1: $step \leftarrow 0$
2: for $epoch$=1 to E do
3:     for $Batch$ from $\{\boldsymbol{X}_{1:n}\}$ do
4:         $\{\boldsymbol{p}_{1:bs}\} \leftarrow RandomCropBlock(Batch)$
5:         $\{\boldsymbol{w}_{1:bs}\}, \{\boldsymbol{b}_{1:bs}\} \leftarrow Rand\_Perm\left(\{\boldsymbol{w}^{(1:B)}_{(step)}\}, bs\right)$
6:         $\theta \leftarrow SGD(\frac{\partial \sum_1^{bs} \mathcal{L}_{\text{lower}}(\boldsymbol{p}_i+\boldsymbol{w}_i,\boldsymbol{b}_i,\theta)}{\partial \theta}, r)$ // Training on classifier
7:         for 1 to 5 do
8:             $\boldsymbol{w}^{(2:B)}_{(step)} \leftarrow Clip_{(-\epsilon,\epsilon)}\left(\boldsymbol{w}^{(2:B)}_{(step)} - \alpha sign(\frac{\partial \sum_1^{bs} \mathcal{L}_{\text{lower}}(\boldsymbol{p}_i+\boldsymbol{w}_i,\boldsymbol{b}_i,\theta)}{\partial \boldsymbol{w}^{(2:B)}_{(step)}})\right)$ // 5-step Projected Gradient Descent
9:         end for
10:        $step \leftarrow step + 1$
11:    end for
12:    return $\{\boldsymbol{w}^{(1:B)}_{(step)}\}$ and $\theta$.
13: end for
```

C. THEORETIC ANALYSIS ON TWO EXAMPLES

In this section, we use two examples, linear regression model and MLP, to show that watermarks with high pattern uniformity can be a stronger feature than others and can be learned easier/earlier than other features. We use MSE as the loss of linear regression and use a general loss in MLP to discuss a general case. We provide the theoretical examples in the two examples to explain that the watermarks with pattern uniformity can be learned prior to other features in the optimization starting at the initialized model.

C.1 Linear regression

Proof of Example 3.5. To reduce the loss by gradient descent, we derive the gradient of L with respect to $\boldsymbol{w}$:

$$
\begin{aligned}
&\mathbb{E}\left[\frac{\partial L}{\partial \mathbf{w}}\right] = \mathbb{E}\left[\frac{\partial(\tilde{\boldsymbol{Z}}^\top \mathbf{w} - \boldsymbol{S}^\top \boldsymbol{\beta} - \boldsymbol{\epsilon})^2}{\partial \mathbf{w}}\right] \\
&= 2\mathbb{E}\left[\tilde{\boldsymbol{Z}}(\tilde{\boldsymbol{Z}}^\top \mathbf{w} - \boldsymbol{S}^\top \boldsymbol{\beta} - \boldsymbol{\epsilon})\right] \\
&= 2\mathbb{E}\left[\tilde{\boldsymbol{Z}}(\tilde{\boldsymbol{Z}}^\top \mathbf{w})\right] - 2\mathbb{E}\left[\tilde{\boldsymbol{Z}}(\boldsymbol{S}^\top \boldsymbol{\beta} + \boldsymbol{\epsilon})\right] \\
&= 2\mathbb{E}\left[(\boldsymbol{MS} + \boldsymbol{\delta})(\boldsymbol{MS} + \boldsymbol{\delta})^\top \mathbf{w}\right] - 2\mathbb{E}\left[(\boldsymbol{MS} + \boldsymbol{\delta})(\boldsymbol{S}^\top \boldsymbol{\beta} + \boldsymbol{\epsilon})\right] \\
&= 2(\mathbb{E}\left[\boldsymbol{MSS}^\top \boldsymbol{M}^\top\right] + \mathbb{E}\left[\boldsymbol{\delta\delta}^\top\right])\mathbf{w} \\
&\quad - 2(\mathbb{E}\left[\boldsymbol{MSS}^\top \boldsymbol{\beta}\right] + \mathbb{E}\left[\boldsymbol{\delta S}^\top \boldsymbol{\beta}\right]) - 2\mathbb{E}\left[(\boldsymbol{MS} + \boldsymbol{\delta})\boldsymbol{\epsilon}\right] \\
&= 2(\mathbb{E}\left[\boldsymbol{MSS}^\top \boldsymbol{M}^\top\right] + \mathbb{E}\left[\boldsymbol{\delta\delta}^\top\right])\mathbf{w} - 2\mathbb{E}\left[\boldsymbol{MSS}^\top \boldsymbol{\beta}\right].
\end{aligned}
\tag{6}
$$

In the above gradient, we separate the hidden feature term according to whether it contains $\mathbf{w}$ to make the comparison with terms with and without $\mathbf{w}$ in watermark term.
For $(\mathbb{E}\left[\boldsymbol{MSS}^\top \boldsymbol{M}^\top\right] + \mathbb{E}\left[\boldsymbol{\delta\delta}^\top\right])\mathbf{w}$, we transform the gradient by $\boldsymbol{M}^\top$ to compare the influence on $\boldsymbol{S}$ by each dimension $\boldsymbol{s}_i$. The norm of the two terms are

$$
\begin{aligned}
\left(\boldsymbol{M}^\top \mathbb{E}\left[\boldsymbol{MSS}^\top \boldsymbol{M}^\top\right] \mathbf{w}\right)_i &= \left(\mathbb{E}\left[\boldsymbol{SS}^\top\right] \boldsymbol{M}^\top \mathbf{w}\right)_i \\
&= \mathcal{O}\left(\frac{1}{d}\left\|\boldsymbol{M}_i^\top \mathbf{w}\right\|\right) \\
&= \mathcal{O}_p\left(\frac{1}{d}\right),
\end{aligned}
\tag{7}
$$

and

$$
\left\|\boldsymbol{M}^\top \mathbb{E}\left[\boldsymbol{\delta\delta}^\top \mathbf{w}\right]\right\| = \left\|\mathbb{E}\left[\boldsymbol{\delta\delta}^\top \mathbf{w}\right]\right\| = \|\mathbb{E}\left[\boldsymbol{\delta}\right]\| \times \mathcal{O}\left(\|\boldsymbol{\delta}\|\right) = \mathcal{O}\left(\|\boldsymbol{\delta}\|^2\right). \tag{8}
$$

When $\|\boldsymbol{\delta}\| \gg 1/\sqrt{d}$, the norm of the watermark term in Eq. 8 is larger than the gradient term from each hidden feature in Eq. 7, which means the watermark feature is learned prior to other hidden features in the first optimization step after model is random initialized.

Similarly, for the rest part in the gradient of Eq. 6, we have

$$\left(\boldsymbol{M}^\top \mathbb{E}\left[\boldsymbol{M}\boldsymbol{S}\boldsymbol{S}^\top\boldsymbol{\beta}\right]\right)_i = \mathcal{O}\left(\frac{1}{d}\left(\boldsymbol{I}_d\boldsymbol{\beta}\right)_i\right) = \mathcal{O}\left(\frac{1}{d}\beta_i\right) = \mathcal{O}\left(\frac{1}{d}\right). \tag{9}$$

When $\|\boldsymbol{\delta}\| \gg 1/\sqrt{d}$, the watermark term in Eq. 8 will have a larger norm than Eq. 9 and the watermark feature can be learned prior to other features.

Combining the other side, when $1/\sqrt{d} \ll \|\boldsymbol{\delta}\| \ll 1/\sqrt{pd}$, because of pattern uniformity, the watermark will have more influence and be learned prior to other hidden features after random initialization even though the watermark has a much smaller norm than each active hidden feature.

On the other hand, assume the watermark $\boldsymbol{\delta}$ has a worse pattern uniformity, and $\boldsymbol{\delta}$ is independent with $\boldsymbol{Z}$. Then the sum of all eigenvalues $\lambda_i(\mathbb{E}[\boldsymbol{\delta}\boldsymbol{\delta}^\top])$ is unchanged, i.e.,

$$\sum_i \lambda_i(\mathbb{E}[\boldsymbol{\delta}\boldsymbol{\delta}^\top]) = tr\left(\mathbb{E}[\boldsymbol{\delta}\boldsymbol{\delta}^\top]\right) = \mathbb{E}tr[\boldsymbol{\delta}\boldsymbol{\delta}^\top] = \mathbb{E}\|\boldsymbol{\delta}\|^2.$$

However, since $\boldsymbol{\delta}$ is random, there are more λ_is which are not zero. Consequently, if we look at the $\left\|\mathbb{E}\left[\boldsymbol{\delta}\boldsymbol{\delta}^\top\mathbf{w}\right]\right\|$, we study

$$\mathbb{E}_{\mathbf{w}}\left\|\mathbb{E}_{\boldsymbol{\delta}}\left[\boldsymbol{\delta}\boldsymbol{\delta}^\top\mathbf{w}\right]\right\|^2 = tr\left(\mathbb{E}[\boldsymbol{\delta}\boldsymbol{\delta}^\top]\mathbb{E}[\boldsymbol{\delta}\boldsymbol{\delta}^\top]\right) = \sum_i \lambda_i(\mathbb{E}[\boldsymbol{\delta}\boldsymbol{\delta}^\top])^2,$$

and then we can find that the average $\left\|\mathbb{E}\left[\boldsymbol{\delta}\boldsymbol{\delta}^\top\mathbf{w}\right]\right\|$ becomes smaller.

On the other hand, it is also easy to figure out that the best $\mathbf{w}$ to minimize L is

$$\mathbf{w}^* = (\boldsymbol{I}_d + \mathbb{E}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top + \boldsymbol{\delta}\boldsymbol{\delta}^\top)^{-1}\boldsymbol{M}\boldsymbol{\beta},$$

i.e., the training process does not forget $\boldsymbol{\delta}$ in the end. □

C.2 Neural Network with a General Task

Remark C.1. While one can obtain a closed-form solution in Example 3.5 for linear regression problem, in Example 3.5, there is no closed-form solution of the trained neural network. Although theoretically tracking the behavior of the neural network is beyond our scope, we highlight that in existing theoretical studies, e.g., [3, 1], the neural network will not forget any learned features during the training.

Proof of Example 3.5. We denote one of the neurons in $\boldsymbol{\mathcal{W}}_1$ as $\mathbf{w}_h$ and shorten the notation of $L\left(\boldsymbol{\mathcal{W}}, \boldsymbol{S}\right)$ as L . In the following, we proof that the gradient updating of each neuron in the first layer is dominated by the $\boldsymbol{\delta}$ because the watermark term has a larger norm compared with other hidden features.

We first derive the gradient of L with respect to $\mathbf{w}_h$:

$$\frac{\partial L}{\partial \mathbf{w}_h} = \frac{\partial L}{\partial \mathbf{w}_h^\top \tilde{\boldsymbol{Z}}}\frac{\partial \mathbf{w}_h^\top \tilde{\boldsymbol{Z}}}{\partial \mathbf{w}_h} = \frac{\partial L}{\partial \mathbf{w}_h^\top \tilde{\boldsymbol{Z}}}\tilde{\boldsymbol{Z}} = \frac{\partial L}{\partial \mathbf{w}_h^\top \tilde{\boldsymbol{Z}}}\left(\boldsymbol{M}\boldsymbol{S} + \boldsymbol{\delta}\right).$$

By denoting $\frac{\partial L}{\partial \mathbf{w}_h^\top \tilde{\boldsymbol{Z}}}$ as $\rho(\tilde{\boldsymbol{Z}})$, we get

$$\frac{\partial L}{\partial \mathbf{w}_h} = \rho(\tilde{\boldsymbol{Z}})\left(\boldsymbol{M}\boldsymbol{S} + \boldsymbol{\delta}\right).$$

For simplicity, we assume $\boldsymbol{M} = \boldsymbol{I}_d$. Then the gradient is

$$\frac{\partial L}{\partial \mathbf{w}_h} = \rho(\tilde{\boldsymbol{Z}})\left(\boldsymbol{S} + \boldsymbol{\delta}\right).$$

To compare the norm of gradient related to $\boldsymbol{x}_i$ with watermark term, we define $\boldsymbol{S}_{-i} = (\boldsymbol{s}_1, ..., \boldsymbol{s}_{i-1}, 0, \boldsymbol{s}_{i+1}, ..., \boldsymbol{s}_d)$, and $\boldsymbol{S}_i = (0, ..., 0, \boldsymbol{s}_i, 0, ..., 0)$. Then

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}_h} &= \rho(\tilde{\boldsymbol{Z}})\left(\boldsymbol{S}_{-i} + \boldsymbol{S}_i + \boldsymbol{\delta}\right)\\
&= \rho(\boldsymbol{S}_{-i} + \boldsymbol{S}_i + \boldsymbol{\delta})\left(\boldsymbol{S}_{-i} + \boldsymbol{S}_i + \boldsymbol{\delta}\right)\\
&= \Bigg[\rho\left(\boldsymbol{S}_{-i}\right) + \rho'\left(\boldsymbol{S}_{-i}\right)^\top\left(\boldsymbol{S}_i + \boldsymbol{\delta}\right) + \frac{1}{2}\left\|\boldsymbol{S}_i + \boldsymbol{\delta}\right\|^2_{\rho''(\boldsymbol{S}_{-i})}\\
&\quad + \mathcal{O}\left(\left\|\boldsymbol{S}_i + \boldsymbol{\delta}\right\|^3\right)\Bigg]\left(\boldsymbol{S}_{-i} + \boldsymbol{S}_i + \boldsymbol{\delta}\right)\\
&= \rho\left(\boldsymbol{S}_{-i}\right)\left(\boldsymbol{S}_{-i} + \boldsymbol{S}_i + \boldsymbol{\delta}\right)\\
&\quad + \rho'\left(\boldsymbol{S}_{-i}\right)^\top\left(\boldsymbol{S}_i + \boldsymbol{\delta}\right)\left(\boldsymbol{S}_{-i} + \boldsymbol{S}_i + \boldsymbol{\delta}\right)\\
&\quad + \frac{1}{2}\left\|\boldsymbol{S}_i + \boldsymbol{\delta}\right\|^2_{\rho''(\boldsymbol{S}_{-i})}\boldsymbol{S}_{-i} + \mathcal{O}\left(\left\|\boldsymbol{S}_i + \boldsymbol{\delta}\right\|^3\right)\\
&= \rho(\boldsymbol{S}_{-i})\boldsymbol{S}_{-i} + \rho'(\boldsymbol{S}_{-i})^\top(\boldsymbol{S}_i + \boldsymbol{\delta})\boldsymbol{S}_{-i}\\
&\quad + \rho(\boldsymbol{S}_{-i})(\boldsymbol{S}_i + \boldsymbol{\delta}) + \rho'(\boldsymbol{S}_{-i})^\top(\boldsymbol{S}_i + \boldsymbol{\delta})(\boldsymbol{S}_i + \boldsymbol{\delta})\\
&\quad + \frac{1}{2}\|\boldsymbol{S}_i + \boldsymbol{\delta}\|^2_{\rho''(\boldsymbol{S}_{-i})}\boldsymbol{S}_{-i} + \mathcal{O}\left(\|\boldsymbol{S}_i + \boldsymbol{\delta}\|^3\right).
\end{aligned}$$

We further assume $\mathbb{E}\rho(\boldsymbol{S}_{-i}) = 0$, $\mathbb{E}\rho'(\boldsymbol{S}_{-i})\boldsymbol{S}_{-i}^\top = 0$, $\mathbb{E}\rho'(\boldsymbol{S}_{-i})^\top\boldsymbol{\delta} = \Theta(\|\boldsymbol{\delta}\|\|\mathbb{E}\rho'(\boldsymbol{S}_{-i})\|)$, and $\|\mathbb{E}\|\boldsymbol{a}\|^2_{\rho''(\boldsymbol{S}_{-i})}\boldsymbol{S}_{-i}\| = \Theta(\|\boldsymbol{a}\|\|\mathbb{E}\rho'(\boldsymbol{S}_{-i})\|)$ for any proper vector $\boldsymbol{a}$[1]. Taking the expectation of the gradient,

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{S}}\left[\frac{\partial L}{\partial \mathbf{w}_h}\right] &= \underbrace{\mathbb{E}\rho(\boldsymbol{S}_{-i})\boldsymbol{S}_{-i}}_{=0} + \underbrace{\mathbb{E}\rho'(\boldsymbol{S}_{-i})^\top\boldsymbol{S}_{-i}(\boldsymbol{S}_i + \boldsymbol{\delta})}_{=0}\\
&\quad + \underbrace{\mathbb{E}\rho(\boldsymbol{S}_{-i})(\boldsymbol{S}_i + \boldsymbol{\delta})}_{=0} + \mathbb{E}\rho'(\boldsymbol{S}_{-i})^\top(\boldsymbol{S}_i + \boldsymbol{\delta})(\boldsymbol{S}_i + \boldsymbol{\delta})\\
&\quad + \mathbb{E}\frac{1}{2}\|\boldsymbol{S}_i + \boldsymbol{\delta}\|^2_{\rho''(\boldsymbol{S}_{-i})}\boldsymbol{S}_{-i} + \underbrace{\mathcal{O}\left(\|\boldsymbol{S}_i + \boldsymbol{\delta}\|^3\right)}_{\text{negligible}}\\
&= \mathbb{E}(\boldsymbol{S}_i + \boldsymbol{\delta})(\boldsymbol{S}_i + \boldsymbol{\delta})^\top\mathbb{E}\rho'(\boldsymbol{S}_{-i})\\
&\quad + \mathbb{E}\frac{1}{2}\|\boldsymbol{S}_i + \boldsymbol{\delta}\|^2_{\rho''(\boldsymbol{S}_{-i})}\boldsymbol{S}_{-i} + o\\
&= \left(\mathbb{E}\boldsymbol{S}_i\boldsymbol{S}_i^\top + \boldsymbol{\delta}\boldsymbol{\delta}^\top\right)\mathbb{E}\rho'(\boldsymbol{S}_{-i})\\
&\quad + \frac{1}{2}\mathbb{E}_{\boldsymbol{S}_{-i}}\left(\mathbb{E}_{\boldsymbol{S}_i}\|\boldsymbol{S}_i\|^2_{\rho''(\boldsymbol{S}_{-i})} + \|\boldsymbol{\delta}\|^2_{\rho''(\boldsymbol{S}_{-i})}\right)\boldsymbol{S}_{-i} + o.
\end{aligned}$$

The notation o represents negligible terms.

Since $\mathbb{E}\rho'(\boldsymbol{S}_{-i})^\top\boldsymbol{\delta} = \Theta(\|\boldsymbol{\delta}\|\|\mathbb{E}\rho'(\boldsymbol{S}_{-i})\|)$, when $\|\boldsymbol{\delta}\| \gg \mathbb{E}\left[\boldsymbol{S}_i\right]$, we have

$$\left\|\left(\mathbb{E}\boldsymbol{S}_i\boldsymbol{S}_i^\top\right)\mathbb{E}\rho'(\boldsymbol{S}_{-i})\right\| \ll \left\|\left(\boldsymbol{\delta}\boldsymbol{\delta}^\top\right)\mathbb{E}\rho'(\boldsymbol{S}_{-i})\right\|.$$

On the other hand, since $\|\mathbb{E}\|\boldsymbol{a}\|^2_{\rho''(\boldsymbol{S}_{-i})}\boldsymbol{S}_{-i}\| = \Theta(\|\boldsymbol{a}\|\|\mathbb{E}\rho'(\boldsymbol{S}_{-i})\|)$, when $\|\boldsymbol{\delta}\| \gg \mathbb{E}\left[\boldsymbol{S}_i\right]$, we have

$$\left\|\mathbb{E}_{\boldsymbol{S}_{-i}}\left(\mathbb{E}_{\boldsymbol{S}_{-i}}\|\boldsymbol{S}_i\|^2_{\rho''(\boldsymbol{S}_{-i})}\right)\boldsymbol{S}_{-i}\right\| \ll \left\|\mathbb{E}_{\boldsymbol{S}_{-i}}\left(\|\boldsymbol{\delta}\|^2_{\rho''(\boldsymbol{S}_{-i})}\right)\boldsymbol{S}_{-i}\right\|.$$

[1] To simplify the analysis, we directly connect $\|\mathbb{E}\|\boldsymbol{a}\|^2_{\rho''(\boldsymbol{S}_{-i})}\boldsymbol{S}_{-i}\|$ to $\|\boldsymbol{a}\|$. To relax this condition, one may consider imposing proper assumptions to exactly derive the formula of $\|\mathbb{E}\|\boldsymbol{a}\|^2_{\rho''(\boldsymbol{S}_{-i})}\boldsymbol{S}_{-i}\|$. We also avoid extreme cases where terms cancel with each other, e.g., $\boldsymbol{\delta}\boldsymbol{\delta}^\top\mathbb{E}\rho'(\boldsymbol{S}_{-i}) = -\mathbb{E}_{\boldsymbol{S}}\|\boldsymbol{\delta}\|^2_{\rho''(\boldsymbol{S}_{-i})}\boldsymbol{S}_{-i}/2$

To summarize, in general, when $\|\boldsymbol{\delta}\| \gg \mathbb{E}[\boldsymbol{S}_i]$, i.e. $\|\boldsymbol{\delta}\| \gg 1/\sqrt{d}$, the norm of the watermark term in the gradient will be much larger than than expectation of any hidden feature, which means the watermark will be learned prior to other features.

The effect of uniformity of $\boldsymbol{\delta}$ follows the same as in Example 3.5.

□

C.3 Experiment to Support Theoretic Analysis with the Two Examples

We use DDPM to learn a watermarked *bird* class in CIFAR10 and compare the accuracy and the quality of generated images in different steps of the training process. The results in Figure 9 show that watermark is much earlier learned before the semantic features, which is consistent with our theoretic analysis in the two examples. In Figure 9, we can see that, at step 20k, the watermark accuracy in generated images is already 94%, but the generated image has no visible feature of bird at all. The bird is generated in high quality until step 60k. This means the watermark is learned much earlier than the semantic features of the images. The observation aligns with our theoretic analysis.

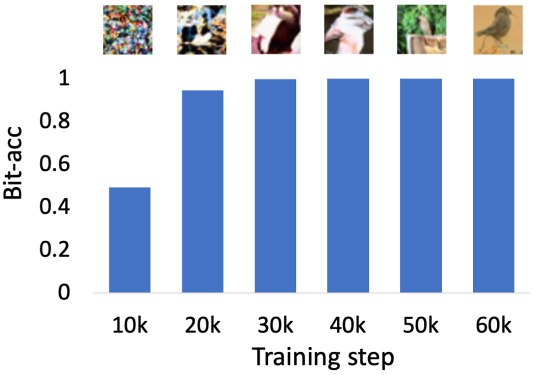


Figure 9: The change of bit accuracy and generated images in the training process.

D. ADDITIONAL DETAILS OF EXPERIMENTAL SETTINGS

D.1 Watermarks and Detector of Experiment for Pattern Uniformity in Section 3.2

In the experiment shown in Figure 3, we test the ability of DDPM [11] to learn watermarks with different pattern uniformity and show more details about the setting in this subsection.

Watermarks. We first choose one class from CIFAR10 as images requiring watermarks $\boldsymbol{X}_{1:R}$, where R is the number of images in this class and $R = 5000$ for CIFAR10. We randomly choose C images from 5 classes from CIFAR10 as $\boldsymbol{W}_{1:R}$, where C is the number of different watermarks and $C = 5, 10, 15, \ldots$. Different watermarks are repeatedly added into $\boldsymbol{X}_{1:R}$ by $\tilde{\boldsymbol{X}}_i = \boldsymbol{X}_i + \sigma \times \boldsymbol{W}_i$. For example, we choose $C = 10$ images as watermarks and every watermark is used to watermark $R/C = 500$ images in $\boldsymbol{X}_{1:R}$. By choosing different C, we can control the uniformity. Larger C means more diverse watermarks and thus smaller pattern uniformity.

Detector. We train a classifier as the detector to detect the watermark in the generated images. The classifier is trained on the images watermarked by 10 classes. The label of the training images is set to be the watermark class. If the classifier predicts that the GDM-generated images have the watermark within the 5 classes from which the C watermarks are chosen, we see it as a successful detection, otherwise it is unsuccessful.

D.2 Block size and message length for different datasets

In our experiment, we considered four datasets, including CIFAR10 and CIFAR100, both with $(U, V) = (32, 32)$, STL10 with $(U, V) = (64, 64)$ and ImageNet-20 with $(U, V) = (256, 256)$. For CIFAR10, CIFAR100 and STL10, we consider the block size (u, v) = (4,4) and B = 4. For ImageNet-20, we set (u,v) = (16, 16) and B = 2. Therefore, for CIFAR10 and CIFAR100, we are able to encode $(\frac{32}{4}) \times (\frac{32}{4}) \times 2 = 128$ bit. For STL-10, we can embed $(\frac{64}{4}) \times (\frac{64}{4}) \times 2 = 512$ bit. And for ImageNet, the message length is $(\frac{256}{16}) \times (\frac{256}{16}) = 256$ bits.

D.3 Decoder Architecture and Details about Training Parameters.

Given the small size of the blocks (4×4), we adapt the original ResNet structure by including only two residual blocks with 64 filters each, positioned between the initial convolutional layer and the global average pooling layer. In the joint optimization, for training decoder, we use the SGD optimizer with momentum to be 0.9, learning rate to be 0.01 and weight decay to be 5×10^{-4}, while for training watermark basic patches, we use 5-step PGD with step size to be 1/10 of the L_∞ budget.

D.4 Details of Baselines

Our method is compared with four existing watermarking methods although they are not specifically designed for the protection of image copyright against GDMs. Information on the baseline methods is provided as follows:

- **Image Blending (IB)**, a simplified version of our approach, which also applies blockwise watermark to achieve pattern uniformity but the patches are not optimized. Instead, it randomly selects some natural images, re-scales their pixel values to 8/255, and uses these as the basic patches. A trained classifier is also required to distinguish which patch is added to a block.

- **DWT-DCT-SVD based watermarking (FRQ)**, one of the traditional watermarking schemes based on the frequency domains of images. It uses Discrete Wavelet Transform (DWT) to decompose the image into different frequency bands, Discrete Cosine Transform (DCT) to separate the high-frequency and low-frequency components of the image, and Singular Value Decomposition (SVD) to embed the watermark by modifying the singular values of the DCT coefficients.

- **HiDDeN** [44], a neural network-based framework for data hiding in images. The model comprises a network architecture that includes an encoding network to hide information in an image, a decoding network to extract the hidden information from the image, and a noise network to attack the system, making the watermark robust. In our main experiments, we did not incorporate noise layers into HiDDeN, except during tests of its robustness to noise (Experiments in 4.7).

- **DeepFake Fingerprint Detection (DFD) [35]**, a method for Deepfake detection and attribution (trace the model source that generated a deepfake). The fingerprint is developed as a unique pattern or signature that a generative model leaves on its outputs. It also employs an encoder and a decoder, both based on Convolutional Neural Networks (CNNs), to carry out the processes of watermark embedding and extraction.

D.5 Details of the Settings of the Corruption Considered in Section 4.7

- Gaussian noise: The mean of the noise is set to 0 the standard deviation is set to 0.1.
- Low-pass Filter: The kernel size of the low-pass filtering is set to 5 and the sigma is 1.
- JPEG Compression: The quality of JPEG Compression is set to 80%.
- Resize: We altered image sizes from 32x32 to 64x64 (termed "large" in the Table 4), or from 32x32 to 16x16 (termed "small" in the table). During detection, we resize all the data back to 32x32 before inputting them to the detector.

D.6 Details of the Experiments about the Generalization to Fine-tuning GDMs

Background in fine-tuning GDMs. To speed up the generation of high-resolution image, Latent Diffusion Model proposes to project the images to a vector in the hidden space by a pre-trained autoencoder [24]. It uses the diffusion model to learn the data distribution in hidden space, and generate images by sampling a hidden vector and project it back to the image space. This model requires large dataset for pre-training and is commonly used for fine-tuning scenarios because of the good performance in pre-trained model and fast training speed of fine-tuning.

Generalization to fine-tuning GDMs. To use our method and enhance the pattern uniformity in the fine-tuning settings, we make two modifications. 1) In stead of enhancing the uniformity in pixel space, we add and optimize the watermark in hidden space and enhance the uniformity in hidden space. 2) Instead of using PGD to limit the budget, we add l_2 norm as a penalty in our objective.

Experiment details. We use the *pokemon-blip-captions* dataset as the protected images and following the default settings in *huggingface/diffusers/examples/text_to_image* [31] to finetune a Stable Diffusion, which is one of Latent Diffusion Models.

E. EXAMPLES OF WATERMARKED IMAGES

Examples of watermarked images are shown in Figure 10, 11 and 12.

F. ADDITIONAL EXPERIMENTAL EVALUATIONS

F.1 Comparison to Additional Baselines

We have conducted a comparison of our method with three other baselines: IGA [36], MBRS [13], and CIN [15]. The results are reported in Table 5 and 6 below. We can see that the performance of IGA is very similar to HiDDeN and DFD. Although IGA's bit accuracy is comparable to our DiffusionShield, it demands a significantly higher budget—more than 20 times the L_∞ and LPIPS values of our approach. As for MBRS and CIN, despite having budgets lower than IGA, they exhibit a worse trade-off between budget and bit accuracy compared to our method, especially on CIFAR100. Specifically, MBRS only attains an 87.68% bit accuracy at twice our budget, and CIN only achieves a 51.13% bit accuracy with a budget close to ours. In contrast, DiffusionShield maintains a high bit accuracy of 99.80% without necessitating a high budget. This is because of the higher pattern uniformity of DiffusionShield. In summary, the performances of theses baselines are similar to the previous baseline methods. They either compromise the budget for bit accuracy close to ours, or fail to be reproduced well in the generated images.

Table 5: Comparison to Additional Baselines with CIFAR-10

Metric	IGA	MBRS	CIN	Ours	
L_∞	52/255	16/255	8/255	**1/255**	2/255
L2	3.38	0.36	0.42	**0.18**	0.36
LPIPS	0.08910	0.00182	0.00185	**0.00005**	0.00020
Cond. Acc.	99.63%	99.97%	99.97%	99.90%	**99.99%**
Uniformity	0.063	0.518	0.599	0.974	0.971

Table 6: Comparison to Additional Baselines with CIFAR-100

Metric	IGA	MBRS	CIN	Ours	
L_∞	66/255	19/255	9/255	**4/255**	8/255
L2	5.31	**0.43**	0.44	0.72	1.43
LPIPS	0.08830	0.00129	**0.00105**	0.00134	0.00672
Cond. Acc.	97.25%	87.68%	51.13%	99.80%	**99.99%**
Uniformity	0.162	0.394	0.527	0.836	0.816

F.2 Robustness under Speeding-up Sampling Models

Speeding-up sampling is often employed by practical GDMs due to the time-consuming nature of the complete sampling process, which requires thousands of steps. However, the quality of the images generated via speeded-up methods, such as Denoising Diffusion Implicit Model (DDIM) [27], is typically lower than normal sampling, which could destroy the watermarks on the generated images. In Table 7, we show the performance of DiffusionShield with DDIM to demonstrate its robustness against speeding-up sampling. Although DiffusionShield has low accuracy on CIFAR100 when the budget is 1/255 and 2/255 (same as the situation in Section 4.2), it can maintain high accuracy on all the other bud-

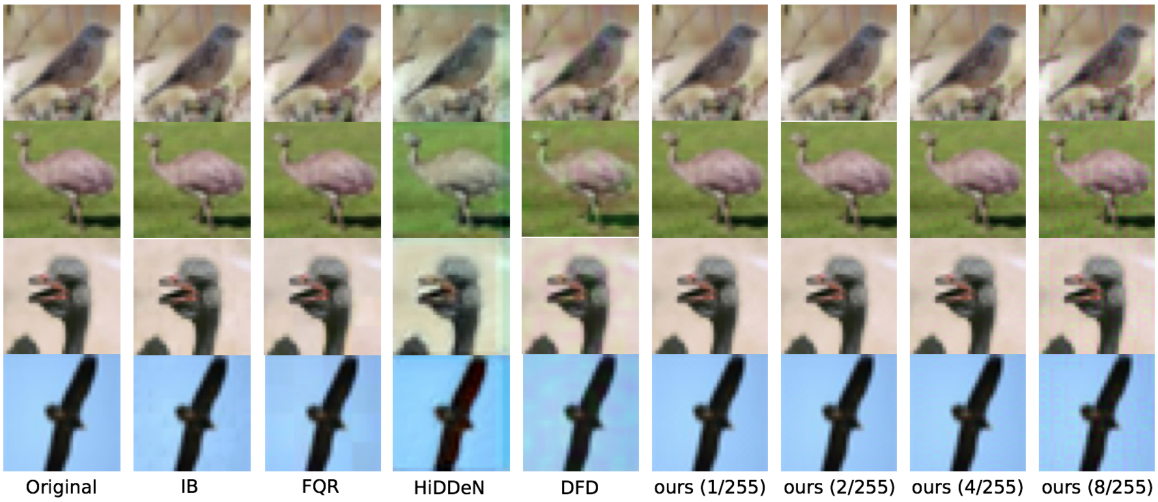


Figure 10: Examples of watermarked images of the bird class in CIFAR-10

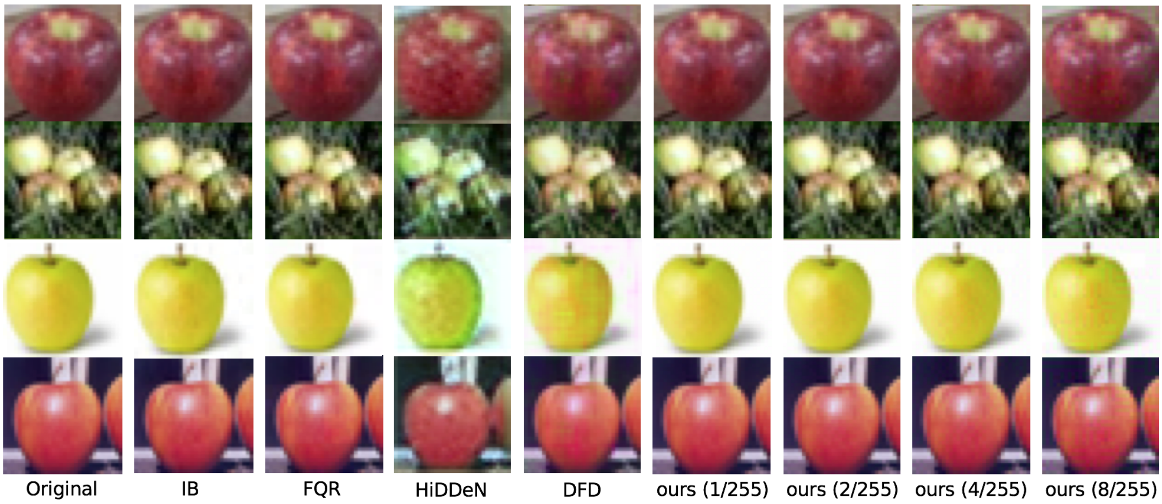


Figure 11: Examples of watermarked images of the apple class in CIFAR-100

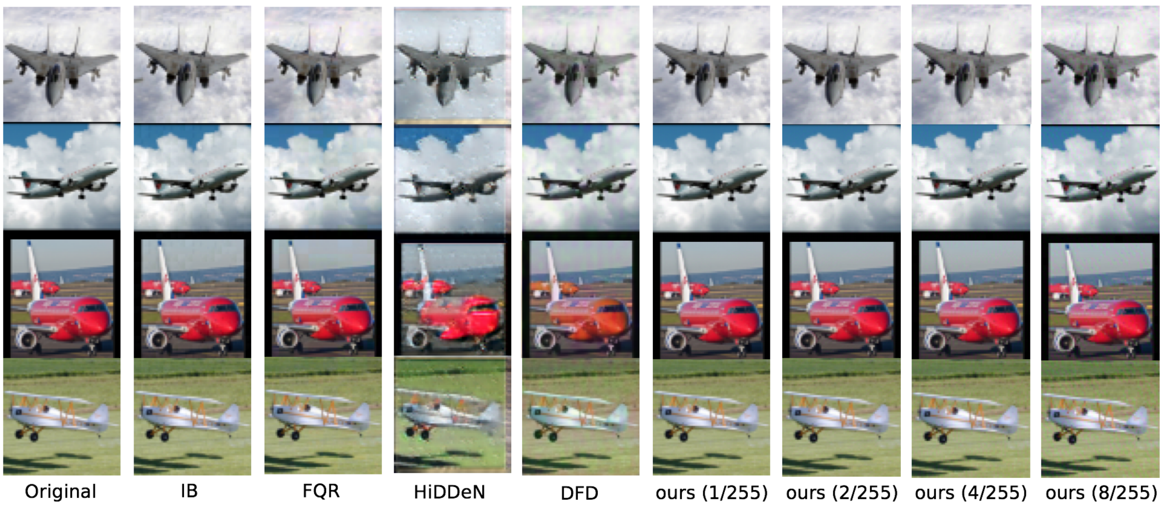


Figure 12: Examples of watermarked images of the plane class in STL-10

Table 7: Bit accuracy (%) with speeding-up models

l_∞		CIFAR10	CIFAR100	STL10
1/255	Cond.	99.7824	52.4813	95.8041
	Uncond.	94.6761	52.2693	82.4564
2/255	Cond.	99.9914	64.5070	99.8299
	Uncond.	96.1927	53.4493	90.4317
4/255	Cond.	99.9996	99.8445	99.9102
	Uncond.	96.1314	92.3109	95.7027
8/255	Cond.	100.0000	99.9984	99.9885
	Uncond.	95.7021	92.2341	95.3009

gets and datasets. Even with a 1/255 l_∞ budget, the accuracy of DiffusionShield on CIFAR10 is still more than 99.7% in class-conditionally generated images and more than 94.6% in unconditionally generated images. This is because the easy-to-learn uniform patterns are learned by GDMs prior to other diverse semantic features like shape and textures. Thus, as long as DDIM can generate images with normal semantic features, our watermark can be reproduced in these images.

F.3 Robustness under Different Hyper-parameters in Training GDMs

Besides the speeded up sampling method, we test two more hyperparameters in Table 8 below. They are learning rate and diffusion noise schedule. Diffusion noise schedule is a hyperparameter that controls how the gaussian noise added into the image increases during the diffusion process. We test with two different schedules, cosine and linear. We use DiffusionShield with 2/255 budget to protect one class in CIFAR10. The results show that the watermark accuracies in all the different parameters are higher than 99.99%, which means our method is robust under different diffusion model hyperparameters.

Table 8: Bit accuracy under different hyper-parameters of DDPM

	cosine	linear
5e-4	99.9985%	99.9954%
1e-4	99.9945%	99.9908%
1e-5	99.9939%	99.9390%

F.4 Watermark's Influence to Generation Quality

In Table 9 and Table 10, we measure the generated quality of both watermarked class and all classes to show that DiffusionShield will not influence the quality of generated images. We use FID to measure the quality of generated images. Lower FID means better generated quality. Comparing FIDs of watermarked classes by different watermark methods, we can find that our method can keep a smaller FID than DFD and HiDDeN when the budget is smaller than 4/255. This means our watermark is more invisible. Comparing FID of ours and clean data, we can find that our method has almost no influence on the generated quality of GDMs. We can also see that FID for the watermarked class is usually higher than FID for all the classes. This is because FID is usually lager when the sample size is small and we sample fewer images in watermarked class than the total number of the samples from all the classes. In summary, our method will not influence the quality of generated images.

Table 9: Generation Quality Measured by FID (only the watermarked class)

method	clean	ours (1/255)	ours (4/255)	ours (8/255)	DFD	HiDDeN
FID	15.633	14.424	26.868	51.027	33.884	48.939

Table 10: Generation Quality Measured by FID (all classes)

method	clean	ours (1/255)	ours (4/255)	ours (8/255)
FID	3.178	4.254	3.926	4.082

G. VISUALIZATION OF FEATURE SPACE

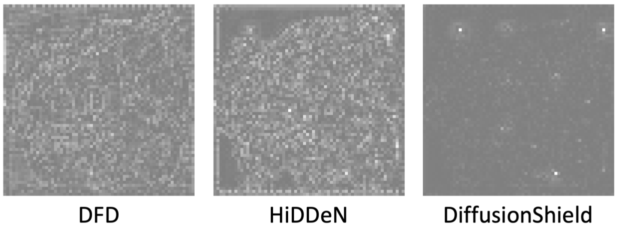


Figure 13: The change of hidden space after watermarking.

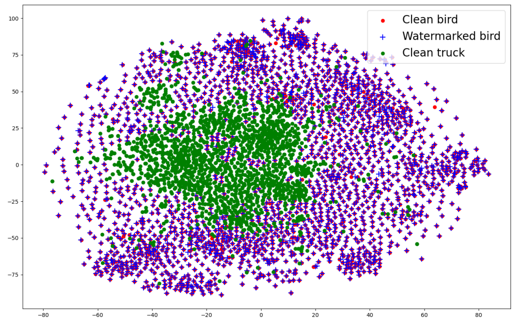


Figure 14: The change of feature space after watermarking.

Visualization of hidden space of Stable Diffusion. In Figure 13, we visualize the change of hidden space. The hidden space of SD is in shape of [4, 64, 64] which has 4 channels. We visualize one of channel and find that the change of DFD and HiDDeN is much obvious than ours.

Visualization of feature space extracted by Contrastive Learning In Figure 14, we visualize the influence of watermark on the feature space. We use Contrastive Learning [4] to extract the feature of both clean and watermarked class.

FT-Shield: A Watermark Against Unauthorized Fine-tuning in Text-to-Image Diffusion Models

Yingqian Cui[1], Jie Ren[1], Yuping Lin[1], Han Xu[2], Pengfei He[1], Yue Xing[1],
Lingjuan Lyu[3], Wenqi Fan[4], Hui Liu[1], Jiliang Tang[1]

[1]Michigan State University [2]The University of Arizona
[3]Sony AI [4]The Hong Kong Polytechnic University

{cuiyingq, renjie3, linyupin, hepengf1, xingyue1, liuhui7, tangjili}@msu.edu
xuhan2@arizona.edu lingjuanlvsmile@gmail.com wenqi.fan@polyu.edu.hk

ABSTRACT

Text-to-image generative models, especially those based on latent diffusion models (LDMs), have demonstrated outstanding ability in generating high-quality and high-resolution images from textual prompts. With this advancement, various fine-tuning methods have been developed to personalize text-to-image models for specific applications such as artistic style adaptation and human face transfer. However, such advancements have raised copyright concerns, especially when the data are used for personalization without authorization. For example, a malicious user can employ fine-tuning techniques to replicate the style of an artist without consent. In light of this concern, we propose FT-Shield, a watermarking solution tailored for the fine-tuning of text-to-image diffusion models. FT-Shield addresses copyright protection challenges by designing new watermark generation and detection strategies. In particular, it introduces an innovative algorithm for watermark generation. It ensures the seamless transfer of watermarks from training images to generated outputs, facilitating the identification of copyrighted material use. To tackle the variability in fine-tuning methods and their impact on watermark detection, FT-Shield integrates a Mixture of Experts (MoE) approach for watermark detection. Comprehensive experiments validate the effectiveness of our proposed FT-Shield.

1. INTRODUCTION

Generative models, particularly Generative Diffusion Models (GDMs) [9, 30, 10, 29], have witnessed significant progress in generating high-quality images from random noise. Recently, text-to-image generative models leveraging latent diffusion [22] have showcased remarkable proficiency in producing specific, detailed images from human language descriptions. Based on this advancement, fine-tuning techniques such as DreamBooth [25] and Textual Inversion [6] have been developed. These methods enable the personalization of text-to-image diffusion models, allowing them to adapt to distinct artistic styles or specific subjects. For instance, with a few paintings from an artist, a model can be fine-tuned to adapt to the artistic style of the artist and create paintings which mimic the style. However, the proliferation of these methods has sparked significant concerns about the potential misuse of these techniques for unauthorized style imitation or the creation of deceptive human facial images. Such actions can potentially violate creators' rights and compromise intellectual property (IP) and privacy integrity [2, 34, 35].

Watermarking has emerged as a popular technique for protecting data's IP against various forms of infringement [3, 21, 18, 36]. It works by injecting imperceptible signals or patterns into images which can later be identified by a watermark detector. This enables the tracking of unauthorized copies and facilitates the assertion of copyright infringement [3, 21, 18, 36]. Compared to other protection methods such as encryption [5], watermarking enjoys several advantages such as its stealthy nature, resilience to manipulation, and the ability to trace and manage digital assets effectively. Given its advantages, the watermarking technique also has the potential for IP protection for text-to-image model fine-tuning. When the watermarked images are used for fine-tuning text-to-image models, the resulting generated images are expected to inherit the watermark, acting as an indelible signature. By employing the detector, the presence of the watermark in the generated content can be identified, providing evidence of IP infringement.

However, we face tremendous challenges in developing watermarking techniques to prevent unauthorized fine-tuning. First, according to prior theoretical findings [1], the fine-tuning process of neural networks involves a sequential pattern in feature learning: certain features can be learned earlier while others are acquired later. This indicates that designing what features to embed within the watermark is crucial. If the watermark's pattern cannot be assimilated by the model prior to style-related features, the infringer can easily circumvent the watermarks by reducing the fine-tuning steps, allowing the model to adopt the style without acquiring the watermark. Although there are recent watermarking methods proposed for images' IP protection against text-to-image model fine-tuning, this challenge has not been solved. As shown in Figure 1, as the fine-tuning steps increase, the watermark proposed by [16] is acquired by the generative model later than the targeted style, indicating an ineffective protection provided by the watermark. One potential reason is that the watermark-generating procedure predominantly adheres to traditional watermark strategies, which are intended to trace the source of an image rather than protecting its IP in the context of diffusion models fine-tuning. Consequently, there's no assurance that the water-

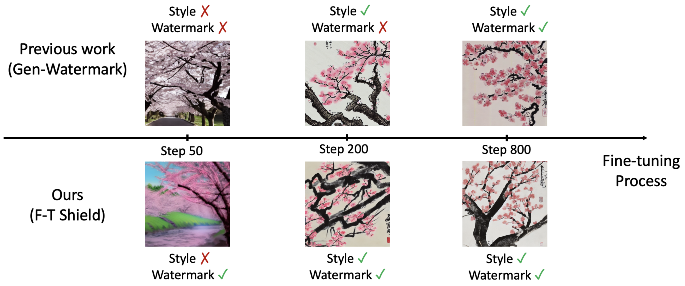


Figure 1: An illustration of generated images from fine-tuned text-to-image models w.r.t. fine-tuning steps. While previous work [16] requires extensive fine-tuning to ensure that the watermark is learned, our method enables the watermark to be learned in the early stages of fine-tuning. Prompt used for the generation: Cherry blossoms in full bloom. Targeted style: the style of artist Beihong Xu.

mark's features will be learned by the model before other features.

Second, as indicated by [16], due to the distribution shift between the fine-tuning data of generative models and the resulting generated images, it is crucial to incorporate images generated by fine-tuned models to develop the watermark detector. Since there are numerous fine-tuning methods introduced from different perspectives, a detector for one method might lose its effectiveness for others. This issue is highlighted in [16] that a large drop in watermark detection performance is observed when applying a detector tailored to one fine-tuning method to others. Meanwhile, in reality, a data protector may not know which fine-tuning method was used by the infringer, thus it is desired to design a watermark detection strategy that is effective for various fine-tuning methods.

To tackle the aforementioned challenges, we propose a novel watermarking framework, **FT-Shield**, tailored for data's copyright protection against the **F**ine-**T**uning of text-to-image diffusion models. In particular, we introduce a training objective incorporating the fine-tuning loss of diffusion models for watermark generation. By minimizing the objective, we ensure that the optimized watermark pattern can be quickly learned by diffusion model at the very early stage of fine-tuning. As shown in Figure 1, even when the style has not been adopted, our watermark has already been learned by the diffusion model. Furthermore, to obtain a watermark detector for various fine-tuning methods, we introduce a detection framework based on Mixture of Experts [12]. The effectiveness of FT-Shield is verified through experiments across various fine-tuning approaches including DreamBooth [25], Textual Inversion [6], Text-to-Image Fine-Tuning [32] and LoRA [11], applied to both style transfer and object transfer tasks across multiple datasets.

2. RELATED WORK

2.1 Text-to-image diffusion model and their fine-tuning methods

Diffusion models [9, 30, 10, 29, 4] have recently achieved remarkable advancements in the realm of image synthesis, notably after the introduction of the Denoising Diffusion Probabilistic Model (DDPM) by [9]. Building upon the DDPM framework, the Latent Diffusion Model (LDM) is introduced in [22]. Unlike conventional models, LDM conducts the diffusion process within a latent space derived from a pre-trained autoencoder, and generates hidden vectors by diffusion process instead of directly generating the image in pixel space. This strategy enables the diffusion model to leverage the robust semantic features and visual patterns imbibed by the encoder. Consequently, LDM has set new standards in both high-resolution image synthesis and text-to-image generation. Building upon text-to-image diffusion models, multiple fine-tuning techniques [6, 25, 11] have been developed. These methods enable the personalization of text-to-image models, allowing them to adapt to distinct artistic styles or specific subjects. Specifically, DreamBooth [6] works by fine-tuning the denoising network of the diffusion model to make the model associate a less frequently used word-embedding with a specific subject. Textual Inversion [25] tries to add a new token which is bound with the new concept to the text-embedding of the model. LoRA [11] adds pairs of rank-decomposition matrices to the existing weights of the denoise network and only trains the newly added weights in fine-tuning.

2.2 Image protection methods

To protect images' IP from unauthorized learning by text-to-image models, in literature, two predominant methods are employed: (1) Adversarial methods which design perturbations in the data to prevent any model learning from the data; and (2) Watermarking techniques which introduce imperceptible signals to the image to enable protectors to detect infringement.

Adversarial methods. Adversarial methods protect data's IP by applying the idea of evasion attacks. They treat the unauthorized generative models as the target of attack, and develop adversarial examples to disrupt the learning process of unauthorized fine-tuning. GLAZE [27] is the first adversarial method which focuses on attacking the features extracted by the encoder in Stable Diffusion to prevent the learning of image styles. The work of [31, 14] introduces methods to generate adversarial examples to evade the infringement from DreamBooth [24] and Textual Inversion [6], respectively. Additionally, it is proposed in [26] to alter the pictures to protect them from image editing applications by Stable Diffusion in case the pictures are used to generate images with illegal or abnormal scenarios. Although these methods provide effective protection against infringement, they can inadvertently disrupt authorized uses (such as for academic research purposes) of the safeguarded images. This indicates the necessity of developing watermarking approaches, which allow the IP to be used for proper reasons while also acting as a way to collect proof against improper uses.

Watermarking methods. Watermark has also been considered to protect the IP of images against unauthorized usage during the fine-tuning of text-to-image models. The work of [33] proposed to apply an existing backdoor method [19] to embed unique signatures into the protected images. It aims to inject extra memorization into the text-to-image models fine-tuned on the protected dataset so that unauthorized data usage can be detected by checking whether the extra memorization exists in the suspected model. However, one limitation is the assumption that the suspicious model is

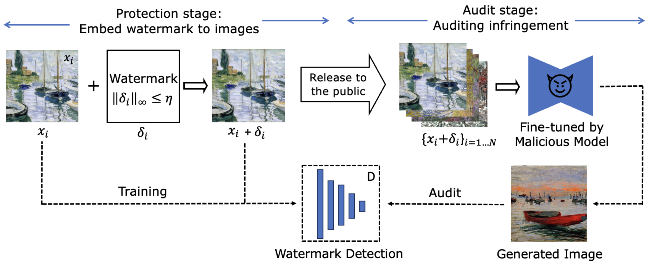


Figure 2: An overview of the two-stage watermarking protection process

readily accessible to the data protector. This might not be realistic, as malicious entities can hide the fine-tuned model and only disclose a handful of generated images. Another work [16] introduces a method that simultaneously trains a watermark generator and detector. The detector is then further fine-tuned with the images generated by the fine-tuned model. However, as discussed in Section 1, the issue of this technique is that it provides no guarantee that their watermark can be learned by the model earlier than the image's style. The shortcomings of current methods highlight the need for a more robust and effective watermarking design.

3. METHOD

In this section, we first define the problem and introduce the key notations. Then we elaborate on some prior theoretical insights to present the challenge in watermark generation and introduce details of the watermark generation process employed by FT-Shield. Lastly, we introduce the MoE framework to detect watermarks on images generated by various fine-tuning methods.

3.1 Problem formulation

In the scenario of copyright infringement and protection considered in this work, there are two roles: (1) a **data protector** that possesses the data copyright, utilizes watermarking techniques before the data is released, and tries to detect if a suspected image is generated by a model fine-tuned on the protected images, and (2) a **data offender** that uses the protected data for text-to-image model fine-tuning without permission from the data protector. The data offenders have complete control over the fine-tuning and sampling processes of the text-to-image diffusion models, while the data protectors can only modify the data they own before their data is released and access images generated by the suspected model.

As shown in Figure 2, the protection process consists of two stages: the protection stage and the audit stage. In the **protection stage**, the data protector protects the images by adding imperceptible watermarks to the images. Specifically, given that the size of the protected dataset is N, the target is to generate sample-wise watermark δ_i for each protected image $x_i, \forall i = 1...N$. Then these watermarks are embedded into the protected images $\hat{x}_i = x_i + \delta_i$. Correspondingly, the data protector develops a watermark detection approach, denoted by function $D_w(\cdot)$, to test whether there is a watermark on the suspected image. To ensure that the watermarks will not lead to severe influence on image quality, we limit the budget of the watermark by constraining its l_∞ norm ($\|\delta_i\|_\infty \leq \eta$) to control the pixel-wise difference between the two images x_i and $\hat{x}_i$. In the **audit stage**, if the protectors encounter suspected images potentially produced through unauthorized text-to-image models fine-tuning, they will apply the watermark detection process $D_w(\cdot)$ to ascertain whether these images have infringed upon their data rights.

3.2 Watermark generation

As mentioned in Section 1, to ensure robust IP protection, it is crucial for the watermarks to be learned earlier in the fine-tuning process. In this subsection, we first introduce the key challenge to achieve this goal by revising how neural networks learn features in pre-training and fine-tuning. Then we propose our watermark generation approach, explaining how it effectively mitigates this challenge.

A key challenge for watermark generation. Based on the theoretical analysis on [1], due to the random initialization, each hidden node in neural networks randomly captures some features at the beginning of training. During pre-training, instead of learning other features from the data, the network emphasizes and strengthens those features that were captured at initialization and are also present in the dataset. Meanwhile, it eliminates features that, despite being learned at initialization, do not find a match in the dataset. Essentially, a well-pre-trained model captures only the features that appear in the pre-training data. An important implication of [1] is that, when learning from the fine-tuning data, the neural network can easily adapt the features that already exist in the pre-training data, but it is difficult to learn new features that only appear in the fine-tuning data. This suggests a challenge: if the watermark's features are new to the diffusion model, it is hard to ensure that the watermark can be easily learned by the model during fine-tuning.

The proposed watermark generation approach. To overcome the above challenge, we propose to simultaneously train the watermark and fine-tune the model as follows. Given N samples to be protected, we construct a training objective for the watermark as:

$$\min_{\theta_1} \min_{\{\delta_i\}_{i\in[N]}} \sum_{i\in[N]} L_{LDM}(\theta_1, \theta_2, x_i + \delta_i, c) \text{ s.t. } \|\delta_i\|_\infty \leq \eta \tag{1}$$

where θ_1 represents the parameters of the UNet [23], which is the denoise model within the text-to-image model structure, θ_2 denotes the parameters of the other part of the diffusion model, and c is the prompt for the image generation. The function L_{LDM} indicates fine-tuning loss of the text-to-image diffusion model:

$$L_{LDM}(\theta_1, \theta_2, x_0, c) = \mathbb{E}_{t,\epsilon\sim\mathcal{N}(0,I_d)} \left\|\epsilon - \epsilon_{\theta_1,\theta_2}(x_t, t, c)\right\|_2, \tag{2}$$

with d as the dimension of the training images in the latent space, x_0 as the input image, t as the timestep and x_t as the input image with t steps' noise in the diffusion process. The above training objective aims to identify the best perturbation δ_i for each sample x_i so that the loss of a diffusion model trained on these perturbed samples can be minimized.

To explain the above design in Eq. 1, since different features are learned differently in the fine-tuning stage, we use a minimization to find the most proper features for the watermark. The inner minimization of δ_i ensures that the watermark

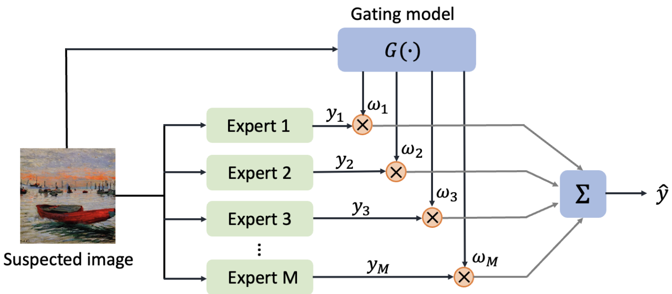


Figure 3: An illustration of mixture of watermark detectors.

can be easily captured by the fine-tuned model, even when the fine-tuning does not involve many steps. Numerically, we solve this bi-level optimization problem by alternatively updating the perturbation and the parameters of the diffusion model. Details about the algorithm are provided in Appendix A.

The design of fine-tuning loss and prompt. There have been multiple fine-tuning methods for text-to-image diffusion models and each of them involves different fine-tuning loss and different parts of the model to be updated. Therefore, it is crucial to ensure that our watermark remains effective across different types of fine-tuning. This requires a careful selection of the specific L_{LDM} in Eq. 1 and the caption c used in watermark training. In terms of the particular L_{LDM} in Eq. 1, to maintain simplicity and coherence, we focused on the most fundamental Text-to-Image Fine-tuning Method [32]. This method aligns with the original training objectives of the text-to-image model and involves updates only to the UNet. Our experiments in Section 4.2 demonstrate that the watermarks can be assimilated well by different fine-tuning methods, even for those which did not modify the UNet such as Textual Inversion [25]. For the caption c, we employ a simplistic and consistent caption format for every image in the dataset in watermark generation. This consistency ensures robustness in varying conditions. Specifically, for the images associated with style transfer tasks, the caption is '*A painting by **' for art datasets, with * denoting the artist's name, and '*A Pokemon character*' for the Pokemon dataset. For images used for object transfer, each is labeled by '*a photo of **', where * indicates the object's category, such as 'toy', or 'person'.

3.3 Mixture of Watermark Detectors

After the watermarks are generated, we need to detect if a suspected image contains a watermark or not. As mentioned in Section 1, a challenge of the watermark detection lies in the problem that the watermark detector tailored to one fine-tuning method cannot transfer well to other methods. This is because different fine-tuning methods modify various parts of the model, resulting in distinct impacts on the generation process and feature updates. To address this issue, we propose a strategy based on the Mixture of Experts (MoE) [12].

The general workflow. An overview of the watermark detection process using MoE is presented in Figure 3. The MoE framework comprises two main elements: the Gating Model and multiple expert models. In the watermark detection phase, the Gating Model is applied to the suspected image to estimate the likelihood that the image was produced by each potential fine-tuning method. Meanwhile, multiple experts of watermark detector, each customized for a distinct fine-tuning method, are employed. The final prediction on the presence of a watermark is then derived by taking a weighted average of these expert assessments, with the weights determined by the Gating Model's probabilities. Formally, the MoE detection framework can be formulated as:

$$D_w(x) = \sum_{i=1}^{M} \text{softmax}_i(G(x)) \cdot E_i(x),$$

where x refers to the image to evaluate, M refers to the total number of expert models, $G(\cdot)$ denotes the Gating Model, $E_i(\cdot)$ represents the i-th expert model, and $\text{softmax}_i(\cdot)$ is the i-th output of the softmax function.

Two-stage training. Instead of training the experts and Gating Model simultaneously, we employ a two-stage training strategy for watermark detection. Specifically, we first train each expert individually with the binary cross-entropy loss. Following the work of [16], we enhance the training dataset for each expert with images generated by fine-tuned text-to-image models. This process unfolds as follows: we first fine-tune the text-to-image model using a particular fine-tuning method with both clean and watermarked datasets, yielding two separately fine-tuned models. Subsequently, these models are employed to generate two distinct sets of data. Data generated from the model fine-tuned with clean data are incorporated into the original clean dataset, and the images generated from the model fine-tuned with watermarked images are utilized to augment the watermarked dataset for the training of the detector. After the experts are developed, we then train the Gating Model with a cross-entropy loss function, leveraging a dataset organized into various classes, where each class contains images generated by a specific fine-tuning method.

This two-stage training strategy has several advantages. First, it provides good adaptability to new fine-tuning methods for text-to-image models. Updating the MoE system requires only the training of the new experts and the Gating Model. This enables the straightforward integration of previously trained experts without necessitating their retraining. Second, it effectively prevents the potential "collapse problem" [28] of MoE, which refers to the situation that the prediction relies on only the output of a single expert. Training the experts separately and initially helps to prevent this issue, promoting a more equitable and effective engagement of all experts. Third, it has a good memory efficiency. Since each expert is trained and inferred independently, it is unnecessary to load all experts into memory at once.

4. EXPERIMENT

In this section, we evaluate the effectiveness of FT-Shield across various fine-tuning methods, subject transfer tasks and different datasets. We first introduce our experimental setups in Section 4.1. In Section 4.2 and 4.3, we evaluate and analyze the detection performance of FT-Shield. Then we assess the impact of FT-Shield on image quality in Section 4.4. We further investigate our approach in Section 4.5 and 4.6 for its performance under fewer fine-tuning steps and robustness against image corruptions. Finally, we conduct ablation studies on FT-Shield's performance with a reduced watermark rate and without training the detector on images from fine-tuned models, whose details are shown in Section 4.7.

4.1 Experimental settings

Model, Task and Dataset. We use Stable Diffusion as the pre-trained text-to-image model. The image size is 512 × 512. We mainly focus on two tasks: style transfer and object transfer. Within style transfer, we explore two subtasks: one centers on art, utilizing 10 diverse datasets from WikiArt, each containing 20 to 40 images; the other focuses on a popular culture, employing the Pokémon BLIP captions dataset [20], which includes 833 images. The object transfer task involves two datasets of lifeless objects from [25] and three datasets of individual human faces from CelebA [15], each comprising five images. We adopt different fine-tuning methods for different tasks. For style transfer, the methods include DreamBooth [25], Textual-inversion [6], Text-to-Image Fine-tuning [32], and LoRA [11], while for object transfer, only DreamBooth and Textual-inversion are utilized because the performance of the other two methods is not satisfying.

Baselines and Scenarios. Our baselines include Gen-Watermark [16] and DIAGNOSIS [33], which are also watermarking methods for IP protection against the fine-tuning of text-to-image model. The two baselines involve different settings in watermark detection. Gen-Watermark [16] requires images generated from fine-tuned models to develop watermark detectors, resulting in the requirement of different watermark detectors tailored to different fine-tuning methods. In contrast, DIAGNOSIS [33] utilizes a general watermark detector to evaluate images generated by different fine-tuning methods. It does not require the knowledge about which specific method is used to generate the suspected image.

To compare with the two baseline methods, we evaluate FT-Shield considering two scenarios: 1) the data protector knows the specific fine-tuning method used by the offender; and 2) the protector is unaware of the fine-tuning method. In the first scenario, the protector can directly apply the watermark detector tailored to that fine-tuning method. We denote our method in this setting as **FT-Shield-Specific** and compare it with Gen-Watermark [16]. In the second case, we employ the watermark detection based on MoE, denoted it as **FT-Shield-MoE**, and compare it with DIAGNOSIS [33].

Implementation Details. For watermark generation, we consider watermark budgets of 4/255 and 2/255 for each dataset. The watermarks are trained with 5-step PGD [17] with the step size to be 1/10 of the budget. For training the experts and Gating Model used in MoE, we adopt ResNet18 [7] with the Adam optimizer [13], taking the learning rate as 0.001 and the weight decay as 0.01. In the detection stage, we use 60 and 30 prompts for image generations in style transfer and object transfer tasks, respectively. Details about the prompts and hyperparameters of the fine-tuning methods are in Appendix E and B.

Evaluation Metrics. We evaluate FT-Shield from two perspectives: 1) its detection performance on data generated by fine-tuned models and 2) its influence on the quality of the released protected images and the generated images. For *Detection Performance*, we consider two metrics. First, the detection rate (or true positive rate, TPR) quantifies the proportion of instances where the detector accurately identifies images produced by models fine-tuned on watermarked images. Second, the false positive rate (FPR) indicates the rate of instances where the detector mistakenly flags images without watermarks as watermarked. For *Image Quality*, we use FID [8] for evaluation. Specifically, we measure the visual discrepancies between the original and watermarked images to evaluate its influence on the released images' quality. We also calculate the FID between the images generated from models fine-tuned on clean images with those generated from models fine-tuned on watermarked images to measure the watermark's influence on the generated images. A lower FID indicates better invisibility.

Table 1: Detection performance of FT-Shield-Specific

		ours ($\eta = 4/255$)		ours ($\eta = 2/255$)		Gen-Watermark[16]	
		TPR↑	FPR↓	TPR↑	FPR↓	TPR↑	FPR↓
Style (Arts)	DreamBooth	**99.50%**	**0.18%**	98.68%	0.87%	93.31%	3.81%
	Textual Inversion	**96.12%**	**3.03%**	93.55%	5.25%	78.75%	12.70%
	Text-to-image	**98.77%**	**1.28%**	96.77%	3.54%	75.41%	30.03%
	LoRA	**97.65%**	**2.67%**	93.37%	6.17%	67.28%	22.72%
Style (Poke-mon)	DreamBooth	**99.5%**	0.50%	98.67%	**0.33%**	95.50%	3.67%
	Textual Inversion	96.00%	**1.67%**	94.67%	4.33%	**97.83%**	2.67%
	Text-to-image	**100.00%**	**0.00%**	99.83%	**0.00%**	98.33%	3.54%
	LoRA	99.67%	**0.00%**	**99.83%**	**0.00%**	97.71%	3.54%
Object	DreamBooth	**98.93%**	1.23%	97.60%	**1.13%**	91.39%	3.50%
	Textual Inversion	**97.73%**	**1.67%**	97.23%	1.97%	88.22%	3.95%

("↑": a higher value is better. "↓": a lower value is better.)

4.2 Effectiveness of FT-Shield-Specific

In this experiment, we evaluate the performance of FT-Shield-Specific. The average of the TPR and FPR across multiple datasets for different transfer tasks are demonstrated in Table 1. According to the results in Table 1, our method is able to protect the images with the highest TPR and lowest FPR among most of the fine-tuning methods in both style and object transfer tasks. With an l_∞ budget of 4/255, the TPR nearly reaches 100% across all fine-tuning methods, while the FPR is close to 0. Even constrained by a small budget (2/255), FT-Shield can still achieve a TPR consistently higher than 90% and an FPR no higher than 7%. In comparison, Gen-Watermark [16] has decent performance in the Pokemon style transfer and object transfer tasks but fails to achieve good performance in the art style transfer task. This indicates that it cannot consistently provide reliable protection across different applications.

Transferability of Tailored Watermark Detector. We explore the transferability of tailored watermark detectors by assessing their performance on images generated by other fine-tuning methods. The performance of FT-Shield-Specific and Gen-Watermark [16] on style transfer tasks is shown in Table 2. The results of object transfer tasks are provided in Appendix D. From Table 2, it can be observed that generally FT-Shield-Specific outperforms Gen-Watermark [16] in terms of the transferability across different fine-tuning methods. Nonetheless, FT-Shield-Specific still experiences an obvious performance drop when the detectors tailored for one fine-tuning method are applied to images generated by other methods. This highlights the need for a method to enhance the adaptability of the detector to different fine-tuning methods. The experiments in Section 4.3 will demonstrate that FT-Shield-MoE can mitigate this problem.

4.3 Effectiveness of FT-Shield-MoE

In this subsection, we demonstrate the effectiveness of FT-

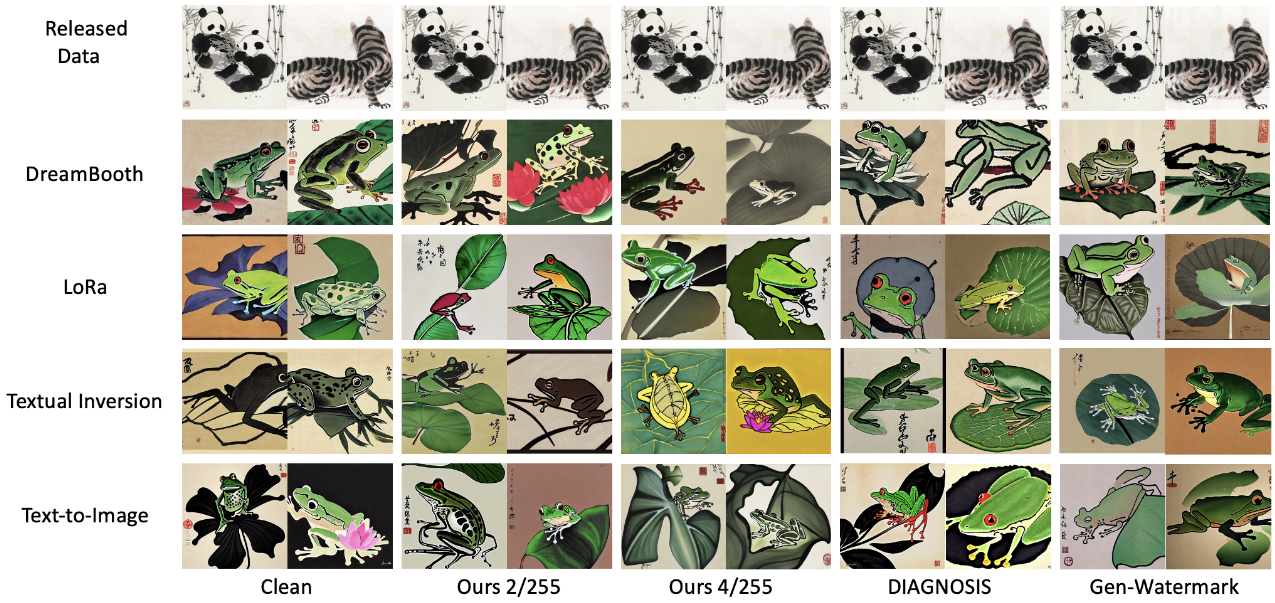


Figure 4: Examples of watermarked images (first line) and generated images (other lines) in the style of artist Beihong Xu. The prompt of generation: A frog on a lotus Leaf.

Table 2: Transferability of the tailored watermark detectors in Style Transfer (Arts). Each number indicates the classifier's detection accuracy (average of true positive rate and true negative rate) when trained on images generated by the column's fine-tuning method and applied to images from the row's fine-tuning method.

Method		Dream-Booth	Textual Inversion	Text-to-Image	LoRA
Ours (2/255)	DreamBooth	98.91%	68.94%	79.29%	78.04%
	Textual Inversion	70.52%	94.15%	57.42%	58.38%
	Text-to-Image	87.27%	57.02%	96.62%	91.15%
	LoRA	91.50%	63.79%	93.19%	93.60%
Ours (4/255)	DreamBooth	99.66%	87.02%	84.38%	83.50%
	Textual Inversion	88.27%	96.55%	68.96%	72.48%
	Text-to-Image	94.35%	68.60%	98.75%	93.73%
	LoRA	97.71%	73.21%	97.54%	97.49%
Gen-Water-mark [16]	DreamBooth	95.92%	64.01%	53.87%	64.11%
	Textual Inversion	68.01%	83.03%	60.23%	75.39%
	Text-to-Image	57.10%	63.94%	72.69%	63.77%
	LoRA	70.75%	66.29%	54.39%	72.28%

Table 3: Performance of FT-Shield-MoE

		ours ($\eta = 4/255$)		ours ($\eta = 2/255$)		DIAGNOSIS[33]	
		TPR↑	FPR↓	TPR↑	FPR↓	TPR↑	FPR↓
Style (Arts)	DreamBooth	**99.42%**	**0.83%**	97.79%	1.83%	84.27%	4.18%
	Textual Inversion	95.37%	0.58%	**95.67%**	3.04%	68.07%	**0.25%**
	Text-to-image	**99.04%**	**1.79%**	96.83%	3.12%	71.87%	5.30%
	LoRA	**97.21%**	**2.87%**	91.08%	6.79%	67.97%	9.78%
Style (Poke-mon)	DreamBooth	**99.00%**	2.17%	98.67%	2.50%	48.11%	**0.67%**
	Textual Inversion	**92.33%**	**1.67%**	91.17%	4.17%	57.05%	4.00%
	Text-to-image	**99.83%**	1.17%	**99.83%**	**0.00%**	82.05%	6.67%
	LoRA	**99.67%**	**0.67%**	**99.67%**	**0.67%**	83.56%	7.83%
Object	DreamBooth	**99.33%**	**1.08%**	98.08%	2.17%	75.97%	1.20%
	Textual Inversion	97.00%	2.08%	**98.17%**	**1.67%**	58.41%	22.20%

Shield-MoE. The detection performance of FT-Shield-MoE when applied to images generated by different fine-tuning methods is shown in Table 3. According to the results, FT-Shield-MoE demonstrates outstanding performance across various datasets. It consistently outperforms the baseline method DIAGNOSIS [33] across different fine-tuning methods and different transfer tasks. For most of the fine-tuning methods, the TPR is consistently higher than 90% and the FPR is consistently lower than 5%, which is comparable to the performance of FT-Shield-Specific under the scenario that fine-tuning method is known to the protector.

4.4 Influence on images quality

In this subsection, we assess how FT-Shield affects the quality of both protected and generated images. Given that the impact of the watermark on image quality is irrelevant to the watermark detection setting, we conduct a collective comparison of FT-Shield with the two baseline methods. We demonstrate the average of the FID metric for each transfer task across different datasets in Table 4. According to the results in Table 4, FT-Shield consistently achieves the lowest FID values in the released dataset. For the generated data, in most cases it also leads to a lighter influence on image quality. Although, in some cases, the FID of images generated by DreamBooth and Textual Inversion is relatively higher, as discussed in Section 4.2 and Section 4.3, FT-Shield consistently achieves higher watermark detection accuracy. This guarantees successful detection of unauthorized usage. To offer a visual perspective, we also provide examples of the watermarked released images and generated images in Figure 4. More visualizations can be found in Appendix C. These visualizations confirm that FT-Shield's watermark is nearly imperceptible, maintaining the aesthetic integrity of both protected and generated images across various fine-tuning models.

4.5 Performance under insufficient fine-tuning steps

In this subsection, we provide more evidence that the watermarks of FT-Shield can be better assimilated by the dif-

Table 4: FID ↓ between clean and watermark images in released (Rel.) and generated (Gen.) images

			FT-Shield (4/255)	FT-Shield (2/255)	Gen-Watermark [16]	DIAGNOSIS [33]
Style (Arts)	Rel.		20.80	**6.79**	58.04	65.50
	Gen.	DreamBooth	62.25	46.96	**46.42**	49.77
		Textual Inversion	67.99	59.25	**41.99**	62.73
		Text-to-image	33.66	**33.00**	38.40	35.71
		LoRA	32.76	**29.99**	30.12	33.30
Style (Pokemon)	Rel.		27.93	**10.63**	57.90	21.03
	Gen.	DreamBooth	43.53	38.14	**32.22**	34.86
		Textual Inversion	96.98	**45.76**	67.24	67.21
		Text-to-image	32.52	**27.22**	47.54	27.73
		LoRA	39.53	**33.52**	49.25	38.82
Object	Rel.		29.45	**10.01**	46.25	57.86
	Gen.	DreamBooth	49.19	41.93	37.57	**37.21**
		Textual Inversion	92.87	**62.67**	102.32	79.58

Table 5: Detection Rate (TPR) under fewer fine-tuning steps

steps	FID	Ours (4/255)	Gen-Watermark [16]	DIAGNOSIS [33]
10	86.05	56.17%	32.67%	3.50%
20	83.90	66.00%	33.50%	7.33%
50	67.32	65.96%	52.67%	2.00%
100	59.42	66.94%	41.00%	1.83%
200	49.84	76.50%	57.97%	13.33%
300	45.93	97.17%	66.67%	54.67%
500	34.66	98.17%	85.67%	79.55%
800	35.25	100.00%	99.72%	93.83%

Table 6: Robustness of the watermark against different image corruptions

Corruption	DreamBooth		Textual Inversion	
Type	w/o aug.	w/ aug.	w/o aug.	w/ aug.
JPEG Comp.	63.83%	99.00%	86.42%	96.58%
Gaussian Noise	68.50%	99.25%	90.17%	97.75%
Gaussian Blur	45.17%	99.25%	75.58%	97.67%
Random Crop	83.83%	99.08%	86.50%	96.50%
Sharpness	98.25%	99.42%	96.92%	98.25%
GreyScale	99.42%	99.42%	93.50%	98.00%
Corruption	Text-to-image		LoRA	
Type	w/o aug.	w/ aug.	w/o aug.	w/ aug.
JPEG Comp.	61.08%	93.67%	79.42%	91.08%
Gaussian Noise	91.08%	91.67%	75.83%	86.17%
Gaussian Blur	92.92%	95.42%	93.08%	91.08%
Random Crop	73.25%	88.00%	71.58%	84.67%
Sharpness	99.92%	100.0%	99.75%	99.83%
GreyScale	99.75%	100.0%	99.33%	99.92%

fusion models at the early stage of the fine-tuning process. Based on DreamBooth, we conduct model fine-tuning with fewer steps compared with standard experiments. Then we apply the watermark detector tailored to DreamBooth to calculate the detection rate (TPR) of the watermark. We also apply FID to evaluate the extent to which the model learns the target style in different steps. The FIDs are derived by comparing images generated at each step against those produced at the completion of fine-tuning. The experiments are mainly done with the style transfer tasks using the paintings of artist Claude Monet. The results are demonstrated in Table 5.

As shown in Table 5, FT-Shield consistently achieves the highest detection rate when the fine-tuning steps are insufficient. Even when the fine-tuning steps are as few as 10, the detection rate can achieve 56%. With only 300 steps, the TPR can achieve nearly 100%. In comparison, the two baseline methods require many more steps to achieve a high detection rate. By observing the change of FID, it can be found that the style has been fully assimilated by the model at the step of 500. At this moment, the TPR of FT-shield is close to 100%, while the other two methods only achieve TPR to be 85.67% and 79.55%. This comparison indicates that our FT-Shield provides more robust copyright protection for data.

4.6 Robustness of watermark

The robustness of a watermark refers to its ability to remain recognizable after undergoing various modifications, distortions, or attacks. It is a crucial property of watermark because during the images' circulation, the watermarks may be distorted by some disturbances, such as JPEG compression. The data offender may also use some methods to remove the watermark. In this subsection, we show that our watermark can be robust against multiple types of corruption when proper augmentation is considered in the training of the watermark detector. In the experiment in Table 6, we consider four types of image corruptions including JPEG compression, Gaussian Noise, Gaussian Blur and Random Crop. To make our watermark robust to those corruptions, we consider using all of these four corruptions as an augmentation in the training of each watermark detector. In Table 6, we show the accuracy of FT-Shield-Specific (average of True Positive Rate and True Negative Rate) which are trained with or without augmentation on the corrupted images. The results corresponding to FT-Shield-MoE are shown in Appendix D. The results indicate that the performance of the watermark detector on the corrupted images is substantially improved after the augmentation is applied during the training of the detector. After the augmentation, the classifier can achieve performance near 100% against all the corruptions in DreamBooth's images. Even in the images generated by LoRA, where the classifier performs the worst, the accuracy can still be consistently higher than 84%.

4.7 Ablation Studies

Reduced watermark rate. Watermark rate refers to the percentage of a dataset that is protected by watermarks. In real practice, the data protector may have already released their unmarked images before the development of the watermark technique. Therefore, it is necessary to consider the situation where the watermark rate is not 100%. In this subsection, we demonstrate the effectiveness of FT-Shield when the watermark rate is lower than 100%. The experiments are mainly based on the style transfer task using the paintings by artist Louise Abbema. The results are presented in Table 7. As shown in the table, as the proportion of the watermarked images in the training set decreases, the TPR also decreases. This is within expectation because when there are fewer watermarked images in the protected dataset, it is harder for the watermark to be assimilated by the diffusion model. Nonetheless, our method consistently achieves better performance than baselines. With a watermark rate of 80%, it achieves a detection rate close to 100% across

Table 7: Watermark detection rate (TPR) under different watermark rates ('FS' and 'FM' refer to 'FT-Shield-Specific' and 'FT-Shield-MoE', respectively, 'GW' stands for 'GenWatermark' [16], and 'DN' represents 'DIAGNOSIS' [33]).

WM Rate	DreamBooth				Textual Inversion			
	FS	GW	FM	DN	FS	GW	FM	DN
100 %	99.66%	98.47%	99.67%	91.50%	96.54%	87.78%	96.83%	88.28%
80 %	99.58%	97.22%	98.83%	90.68%	97.75%	84.86%	95.00%	88.35%
50 %	95.92%	94.45%	87.33%	54.91%	92.92%	83.20%	88.50%	37.60%
20 %	86.42%	92.78%	83.50%	13.81%	88.25%	81.25%	85.83%	19.30%

WM Rate	Text-to-image				LoRA			
	FS	GW	FM	DN	FS	GW	FM	DN
100 %	98.74%	93.33%	99.33%	88.89%	97.49%	85.83%	98.33%	87.83%
80 %	96.92%	80.70%	96.00%	63.73%	95.00%	78.34%	91.67%	75.21%
50 %	86.25%	77.50%	83.67%	48.75%	73.75%	66.11%	66.17%	64.39%
20 %	83.33%	74.45%	81.00%	32.11%	63.83%	54.45%	56.50%	44.76%

Table 8: Performance of watermark detector trained without augmentation

		Ours (4/255)	Ours (2/255)
DreamBooth	TPR↑	84.67%	79.06%
	FPR↓	9.56%	9.61%
Textual Inversion	TPR↑	54.83%	47.61%
	FPR↓	3.83%	13.22%
Text-to-image	TPR↑	37.89%	46.00%
	FPR↓	2.89%	15.50%
LoRA	TPR↑	44.06%	53.61%
	FPR↓	4.72%	19.67%

all fine-tuning methods. Even when the watermark rate is reduced to 20%, FT-Shield still maintains detection rates higher than 80% across all the fine-tuning methods except LoRA. Although the watermark detection rate for LoRA's generated images experienced the most substantial decline, it remains much higher than the two baseline methods.

Detector trained without data augmentation. As discussed in Section 3.3, it is necessary to use the generated data from the fine-tuned diffusion model to augment the dataset used for the watermark detector's training. Table 8 demonstrates the performance of the classifier if there is no augmentation (based on the style transfer task). The classifier is simply trained on the dataset which contains the clean and watermarked protected images. According to the results demonstrated in Table 8, when there is no augmentation, the watermark detector can successfully detect some watermarked images on the generated set, especially those generated by DreamBooth. However, the performance will be much worse than the ones with augmented data. This difference demonstrates the necessity to conduct augmentation when training the watermark detector.

5. CONCLUSION

In this paper, we proposed a novel watermarking method to safeguard images' IP against the fine-tuning of text-to-image diffusion models. To ensure that the watermark can be efficiently and accurately assimilated by the diffusion model, we proposed an algorithm for watermark generation which incorporates the fine-tuning loss of diffusion models in the training loss of watermark. Meanwhile, we introduce a MoE strategy for the watermark detection to enhance its adaptability to diverse fine-tuning methods. Empirical results demonstrates the effectiveness of our method and its superiority over the existing watermarking methods.

References

[1] Zeyuan Allen-Zhu and Yuanzhi Li. Feature purification: How adversarial training performs robust deep learning. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 977–988. IEEE, 2022.

[2] Chen Chen, Jie Fu, and Lingjuan Lyu. A pathway towards responsible ai generated content. *arXiv preprint arXiv:2303.01325*, 2023.

[3] Ingemar Cox, Matthew Miller, Jeffrey Bloom, and Chris Honsinger. Digital watermarking. *Journal of Electronic Imaging*, 11(3):414–414, 2002.

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[5] Behrouz A Forouzan. *Cryptography & network security.* McGraw-Hill, Inc., 2007.

[6] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[12] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, XUE Zhengui, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. 2023.

[15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[16] Yihan Ma, Zhengyu Zhao, Xinlei He, Zheng Li, Michael Backes, and Yang Zhang. Generative watermarking against unauthorized subject-driven image synthesis. *arXiv preprint arXiv:2306.07754*, 2023.

[17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the international conference on learning representations*, 2018.

[18] KA Navas, Mathews Cheriyan Ajay, M Lekshmi, Tampy S Archana, and M Sasikumar. Dwt-dct-svd based watermarking. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, pages 271–274. IEEE, 2008.

[19] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021.

[20] Justin N. M. Pinkney. Pokemon blip captions. `https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions/`, 2022.

[21] Christine I Podilchuk and Edward J Delp. Digital watermarking: algorithms and applications. *IEEE signal processing Magazine*, 18(4):33–46, 2001.

[22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[24] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022.

[25] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[26] Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. *arXiv preprint arXiv:2302.06588*, 2023.

[27] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by text-to-image models. *arXiv preprint arXiv:2302.04222*, 2023.

[28] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[30] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[31] Thanh Van Le, Hao Phung, Thuan Hoang Nguyen, Quan Dao, Ngoc Tran, and Anh Tran. Anti-dreambooth: Protecting users from personalized text-to-image synthesis. *arXiv preprint arXiv:2303.15433*, 2023.

[32] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers/tree/main/examples/text_to_image`, 2022.

[33] Zhenting Wang, Chen Chen, Lingjuan Lyu, Dimitris Metaxas, and Shiqing Ma. Diagnosis: Detecting unauthorized data usages in text-to-image diffusion models. In *International Conference on Learning Representations*, 2024.

[34] Zhenting Wang, Chen Chen, Yi Zeng, Lingjuan Lyu, and Shiqing Ma. Where did i come from? origin attribution of ai-generated images. *Advances in Neural Information Processing Systems*, 36, 2024.

[35] Yuxin Wen, Yuchen Liu, Chen Chen, and Lingjuan Lyu. Detecting, explaining, and mitigating memorization in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023.

[36] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 657–672, 2018.

APPENDIX

A. ALGORITHM

The detailed algorithm for the training of the watermark (Eq. 1) is demonstrated as below.

Algorithm 1 Optimization for watermark δ_i

```
Input: Protected dataset $\{x_i\}_{i\in[n]}$, Captions for protected dataset $\{c_i\}_{i\in[n]}$, Initialized watermark $\{\delta_{i,0}\}_{i\in[n]}$, Pre-trained text-to-image diffusion model with parameters $\theta_1, \theta_2$ ($\theta_1$ denotes the unet part and $\theta_2$ denotes the other parts), watermark budget $\eta$, diffusion model learning rate $r$, batch size $bs$, PGD step $\alpha$ and epoch $E$
Output: Optimal watermark $\{\delta_i^*\}_{i\in[n]}$
for Epoch=1 to E do
    for Batch from $\{x_i\}_{i\in[n]}$ do
        $\theta_1^* \leftarrow \theta_1$
        for 1 to 5 do
            $\theta_1^* \leftarrow \theta_1^* - r\frac{\partial}{\partial\theta_1^*}L_{dm}(\theta_1^*, \theta_2, x_{1:bs}, c_{1:bs})$ // Use clean images to update the unet
        end for
        for 1 to 5 do
            $\delta_{1:bs} \leftarrow \delta_{1:bs} - \alpha sign\{\frac{\partial}{\partial\delta_{1:bs}}L_{dm}(\theta_1^*, \theta_2, x_{1:bs} + \delta_{1:bs}, c_{1:bs})\}$
            $\delta_{1:bs} \leftarrow Proj_{\|\delta_{1:bs}\|_\infty\leq\eta}(\delta_{1:bs})$ // PGD to update watermark
        end for
        for 1 to 5 do
            $\theta_1 \leftarrow \theta_1 - r\frac{\partial}{\partial\delta_{1:bs}}L_{dm}(\theta_1, \theta_2, x_{1:bs} + \delta_{1:bs}, c_{1:bs})$ // Use watermarked images to update the unet
        end for
    end for
end for
```

B. ADDITIONAL DETAILS ABOUT THE FINE-TUNING METHODS

In the experiments of this paper, we considered four Fine-tuning methods of text-to-image models including DreamBooth, Textual Inversion, Text-to-Image and Text-to-Image-LoRA for style transfer and object transfer tasks. More details about the setting of these fine-tuning methods are provided as below.

- **DreamBooth** [25]: DreamBooth is a fine-tuning method to personalize text-to-image diffusion models. It mainly focus on fine-tuning the unet of the diffusion models' structure with L_{LDM} adding a prior preservation loss to avoid overfitting and language-drift. Whether to update the text-encoder within in the text-to-image model structure is an open option. In the experiment in this paper, we update both the unet and the text-encoder with learning rate to be 2e-6, batch size to be 1 and maximum fine-tuning stpes to be 800. For style transfer task, we use "[V]" as the unique identifier for the specific style and incorporate "[V]" in the prompts in the sampling process to instruct the model to generated images following this style. Similarly, for object transfer, we use "sks" as the identifier.

- **Textual Inversion** [6]: Textual Inversion is another text-to-image diffusion model personalization method. It focuses on adding a new token which is connected to a specific style or object to the vocabulary of the text-to-image models. This work by using a few representative images to fine-tune the text embedding of the pipeline's text-encoder. The loss of the fine-tuning is L_{LDM}. In our experiment, we set the fine-tuning learning rate to be 5.0e-04, batch size to be 1 and maximum fine-tuning steps to be 1500. We use "[V]" as a placeholder to represent the new concepts that the fine-tuning process learn and also incorporate it in the prompts in the sampling process.

- **Text-to-Image**: Text-to-Image Fine-Tuning Method is a simple implementation of the fine-tuning of text-to-image diffusion models. It uses L_{LDM} as objective and fine-tunes the whole unet structure with a dataset which contains both the images and the captions describing the contents of the images. An identifier "[V]" is also required in the captions in the fine-tuning and sampling procedure. In our experiment, we set the learning rate for fine-tuning to be 5e-06, the batchsize to be 6 and the maximum fine-tuning steps to be 300.

- **LoRA** [11]: LoRA works by adding pairs of rank-decomposition matrices to existing weights of the UNet and only train the newly added weights in the fine-tuning process (also using L_{LDM} as objective). It also required a image-caption pair dataset for fine-tuning and and the identifier "[V]" in the cations. In experiments, we set the learning rate to 5e-06, batch size to 6 and maximum training steps to 3000.

C. ADDITIONAL VISUALIZATION OF WATERMARKS

In this section, we provide some additional visualization of watermarks as in Figure 5, 6, and 7.

D. ADDITIONAL EMPIRICAL RESULTS

Transferability of tailored watermark detector. We provide some additional results (results for the object transfer tasks) regarding the transferability of tailored watermark detector here. The results are shown in Table 2, indicating that the performance of the detectors is greatly reduced when the classifier is applied to the images generated by a different fine-tuning method.

Robustness of watermark (FT-Shield-MoE). Given the similar performance of FT-Shield-MoE to FT-Shield-Specific, in Section 4.6 we primarily present the detection performance of FT-Shield-Specific. Nonetheless, detection performance of FT-Shield-MoE is also provided, as seen in Table 10. Comparing Table 6 and Table 10 we can see that, in general, the detection performance of FT-Shield-MoE is similar to that of FT-Shield-Specific. Both FT-Shield-Specific and FT-Shield-MoE achieve good performance in watermark detection on corrupted generated images.

E. PROMPTS USED FOR IMAGES GENERATION

In the following, we provide the prompts used in image generation in our experiments.

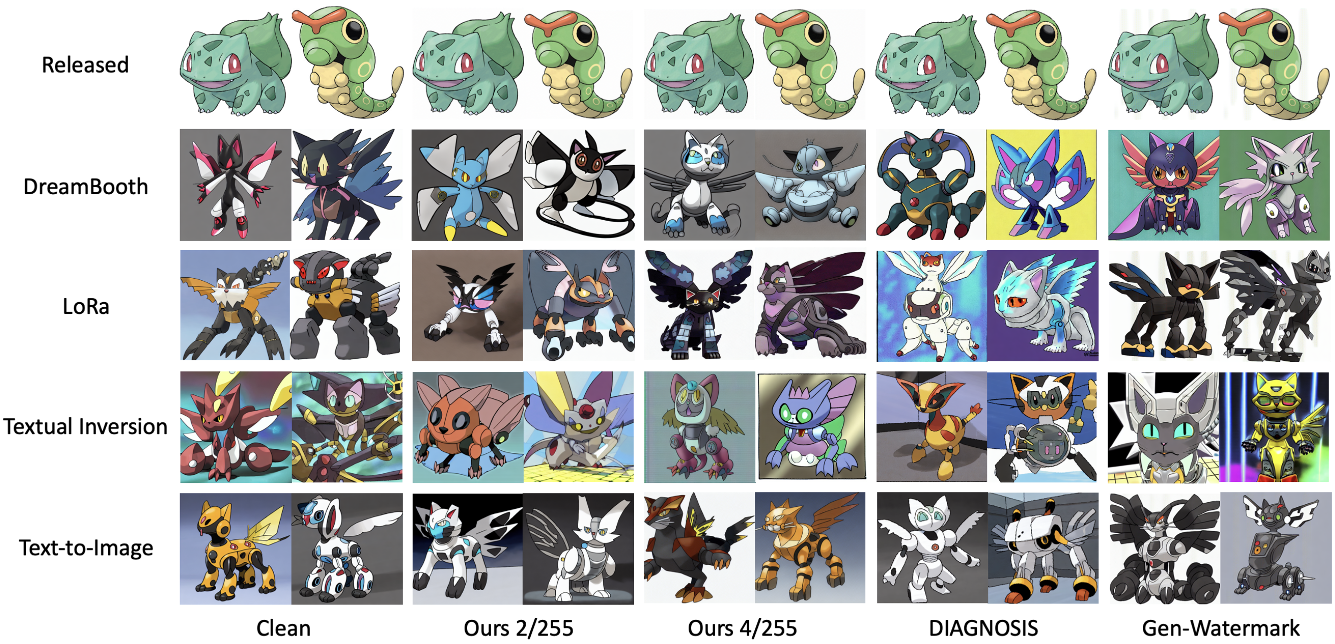


Figure 5: Examples of watermarked images (first line) and images generated through domain adaptation for Pokemon imagery (other lines). The prompt of generation: A robotic cat with wings.

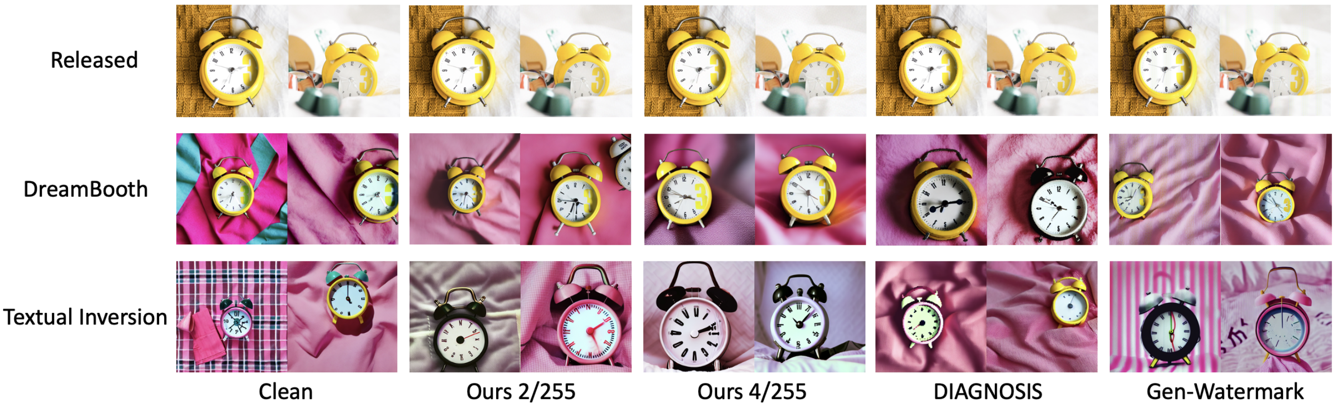


Figure 6: Examples of watermarked images (top row) and images generated from object transfer (clock) shown in subsequent rows. The prompt of generation: A sks clock on top of pink fabric.

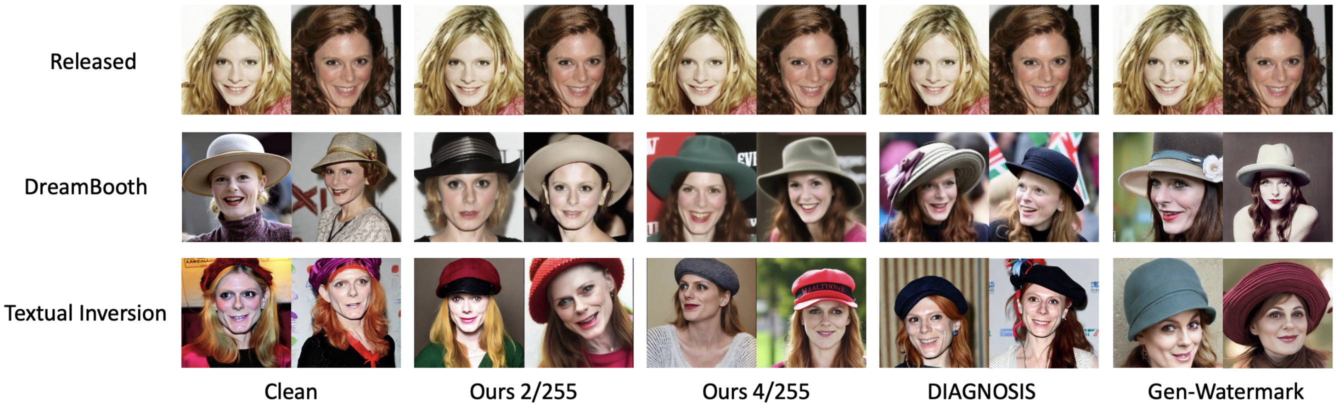


Figure 7: Examples of watermarked images (first line) and images generated from face transfer (other lines). The prompt of generation: A photo of sks person wearing a vintage hat.

Table 9: Transferability of the watermark detectors in Style Transfer (Object)

budget		DreamBooth	Textual Inversion
2/255	DreamBooth	98.24%	57.28%
	Textual Inversion	68.23%	97.63%
4/255	DreamBooth	98.85%	73.77%
	Textual Inversion	72.05%	98.03%

Table 10: Robustness of the watermark against different image corruptions (detected by FT-Shield-MoE)

Corruption	DreamBooth		Textual Inversion	
Type	w/o aug.	w/ aug.	w/o aug.	w/ aug.
JPEG Comp.	62.17%	98.92%	75.42%	98.33%
Gaussian Noise	84.58%	98.67%	95.00%	96.67%
Gaussian Blur	89.75%	99.17%	69.58%	95.42%
Random Crop	82.83%	98.83%	73.75%	95.83%
Corruption	Text-to-image		LoRA	
Type	w/o aug.	w/ aug.	w/o aug.	w/ aug.
JPEG Comp.	85.33%	94.33%	76.17%	90.08%
Gaussian Noise	71.33%	88.42%	78.33%	85.42%
Gaussian Blur	92.17%	94.00%	89.03%	91.50%
Random Crop	71.92%	84.08%	73.67%	86.92%

E.1 Prompts Used for Face Object Transfer

A photo of [V] laughing heartily
A photo of [V] by the beach at sunset
A photo of [V] wearing a vintage hat
A photo of [V] with a glass of wine
A photo of [V] reading a thick book
A photo of [V] wearing a graduation cap
A photo of [V] with ear rings
A photo of [V] holding a vintage camera
A photo of [V] holding a coffee cup
A photo of [V] holding a bouquet of flowers
A photo of [V] in a classroom
A photo of [V] wearing A winter scarf and gloves
A photo of [V] on a boat
A photo of [V] wearing oversized sunglasses
A photo of [V] in the jungle
A photo of [V] in front of a flower field
A photo of [V] with a kitten
A photo of [V] amidst colorful autumn leaves
A photo of [V] in a sunny park
A photo of [V] holding a bottle of water
A photo of [V] in her bedroom
A photo of [V] surrounded by festive balloons
A photo of [V] with upset face
A photo of [V] with a colorful parrot on the shoulder
A photo of [V] with blunt-cut bangs
A photo of [V] with straight black hair
A photo of [V] in the street
A photo of [V] in front of a window
A photo of [V] with short hair
A photo of [V] in a library, surrounded by towering bookshelves

E.2 Prompts Used for Lifeless Object Transfer

A [V] in the snow
A [V] with a wheat field in the background
A [V] on the beach
A [V] with a tree and autumn leaves in the background
A [V] on a cobblestone street
A [V] with the Eiffel Tower in the background
A [V] on top of pink fabric
A [V] on top of green grass with sunflowers around it
A [V] on top of a wooden floor
A [V] on top of a mirror
A [V] with a city in the background
A [V] on top of a dirt road
A man with a [V]
A [V] on top of a white rug
A red [V]
A [V] with a blue house in the background
A cube shaped [V]
A [V] placed beside a window
A girl holding a [V]
A [V] on a desk
A [V] on a chair
A [V] beside a computer
A [V] on the top of a roof
A [V] in a box
A [V] on a bed
A [V] with a mountain in the background
A [V] on a desk
A [V] on a cliff overlooking the sea
A [V] placed on a bookshelf
A [V] under a tree

E.3 Prompts Used for Style Transfer (Arts)

A lady reading on grass in the style of [V]
Anglers on the Seine River in the style of [V]
Flower field in the style of [V]
Haystacks in winter mornings in the style of [V]
Iris in the style of [V]
Mother and her child in a garden in the style of [V]
Red Boats at Argenteuil in the style of [V]
Saint Lazar Railway Station in the style of [V]
Sunflowers in the style of [V]
A man in a suit with a beard in the style of [V]
Waterfall in the style of [V]
Boats at rest at petit gennevilliers in the style of [V]
Eese in the creek in the style of [V]
Claude haystack at giverny in the style of [V]
Meadow with poplars in the style of [V]
Olive tree wood in the moreno garden in the style of [V]
A fountain in the style of [V]
A bottle of champagne in an ice bucket in the style of [V]
Snow scene in the style of [V]
An Italian vineyard at midday in the style of [V]
The artist house in the style of [V]
Cherry blossoms in full bloom in the style of [V]
The cabin in the style of [V]
The bodmer oak fontainebleau in the style of [V]
Sunrise in the style of [V]
The sea at saint adresse in the style of [V]
The seine in the style of [V]
Birds taking flight from a tree in the style of [V]
Walk in the meadows in the style of [V]
The summer poppy field in the style of [V]
Rough sea in the style of [V]
An old man playing the violin in the style of [V]
Woman in a garden in the style of [V]
Swans gliding on a serene pond in the style of [V]
A growling tiger in the style of [V]
Majestic castle overlooking a valley in the style of [V]

Waterloo bridge in the style of [V]
Wild horses galloping on the shore in the style of [V]
The boat studio in the style of [V]
Bustling train station in the 1900s in the style of [V]
The sheltered path in the style of [V]
Portrait of a Woman with Low Neckline in the style of [V]
Ballerinas rehearsing in the style of [V]
Market day in a provincial town in the style of [V]
A goat on grass in the style of [V]
The old lighthouse by the cliff in the style of [V]
Two butterflys in the style of [V]
Moonlit night over a calm sea in the style of [V]
A frog on a lotus Leaf in the style of [V]
A camel team in the desert in the style of [V]
An eggplant on vines in the style of [V]
Windmills on the Flower Field in the style of [V]
Alm tree at bordighera in the style of [V]
A dog waiting for the owner in the style of [V]
A view of mountain in the style of [V]
Two pandas eating bamboos in the style of [V]
Grapes on a vine in the style of [V]
A woman admiring lotus flowers in the style of [V]
The Cliffs of Etelta in the style of [V]
The side face of a red-haired woman in the style of [V]

Remarks: In the sampling procedure of DreamBooth and Textual Inversion, the prompts need to be added with the word ”painting”, e.g., A lady reading on grass in the style of [V] painting.

E.4 Prompts Used for Style Transfer (Pokemon)

A robotic cat with wings
A blue and white dinosaur with wings
A phoenix with icy feathers
A cartoon sunflower with a happy face
A steel-spined hedgehog
A deer with colorful feathers on it’s head
A small, furry creature with large eyes
A gray cartoon character with a black tail
A fire-breathing fox
A pokemon ball with a butterfly on top of it
A jellyfish-like creature
A red and white toy with a blinking green eye
A robotic unicorn with a laser horn
A small, insect-like creature with petal-like wings
A robotic dog with butterfly wings
A magical gnome that walks at night
A tree-like creature with glowing eyes
A bioluminescent jellyfish with a galaxy pattern inside
A creature made of clouds
A cartoon character with claws
A creature with butterfly-like wings
A yellow and orange pokemon with big red eyes
A fairy-tale dragon
A miniature dragon made of living crystal
An armored turtle with a spiked tail
A will-o’-the-wisp in the form of a playful kitten
A cyborg rabbit with blue eyes
A floating island turtle with a mini ecosystem on its back
A robotic owl with holographic wings
A dog with a wheel in his hand
A cyborg penguin with jet engines
A mechanical elephant with hover discs for feet
A robot bear with solar panels as fur
A spectral wolf with aurora-like fur
A blue and yellow insect
An origami crane that comes to life
A pink dog with red ears sitting down
A chameleon with digital camouflage
A goat with a bell on its head
A peacock with holographic tail feathers
A red and white dragonfly
A saber-toothed cat with plasma claws
A blue spider with a green cap
A clockwork bird with wings of stained glass
A green and yellow bear
A starry narwhal with a radiant horn
A green and red dragon
An ice-cream shaped panda with flavors as fur colors
A red and orange pokemon
A shadow fox that morphs into smoke
A cute animal with horns
A wind-up mouse with antique clockwork gears
A colorful butterfly
A coral reef mermaid with sea glass and shells
A pair of owls with orange wings
A robot with a green body, yellow arms, and a red head
A bamboo dragon that grows real leaves
A glowing, ethereal deer with constellation patterns in its fur
A neon-furred squirrel with jetpack wings, channeling the futuristic vibe
An aquatic creature with the features of a shark and a dolphin

Graph Fairness via Authentic Counterfactuals: Tackling Structural and Causal Challenges

Zichong Wang[1], Zhipeng Yin[1], Fang Liu[2], Zhen Liu[3], Christine Lisetti[1], Rui Yu[4],
Shaowei Wang[5], Jun Liu[6], Sukumar Ganapati[1], Shuigeng Zhou[7], and Wenbin Zhang[1*]
[1] Florida International University, Miami, FL, USA
[2] University of Notre Dame, Notre Dame, IN, USA
[3] Guangdong University of Foreign Studies, Guangzhou, China
[4] University of Louisville, Louisville, KY, USA
[5] University of Manitoba, Winnipeg, Manitoba, Canada
[6] Carnegie Mellon University, Pittsburgh, PA, USA
[7] Fudan University, Shanghai, China

ABSTRACT

The extensive use of graph-based Machine Learning (ML) decision-making systems has raised numerous concerns about their potential discrimination, especially in domains with high societal impact. Various fair graph methods have thus been proposed, primarily relying on statistical fairness notions that emphasize sensitive attributes as a primary source of bias, leaving other sources of bias inadequately addressed. Existing works employ counterfactual fairness to tackle this issue from a causal perspective. However, these approaches suffer from two key limitations: they overlook hidden confounders that may affect node features and graph structure, leading to an oversimplification of causality and the inability to generate authentic counterfactual instances; they neglect graph structure bias, resulting in over-correlation of sensitive attributes with node representations. In response, this paper introduces the *Authentic Graph Counterfactual Generator (AGCG)*, a novel framework designed to mitigate graph structure bias through a novel fair message passing technique and to improve counterfactual sample generation by inferring hidden confounders. Comprising four key modules – subgraph selection, fair node aggregation, hidden confounder identification, and counterfactual instance generation – AGCG offers a holistic approach to advancing graph model fairness in multiple dimensions. Empirical studies conducted on both real and synthetic datasets demonstrate the effectiveness and utility of AGCG in promoting fair graph-based decision-making.

1. INTRODUCTION

Graph data is prevalent in real-world scenarios, such as financial markets [54], item recommendations [49], and social networks [35]. Distinguished from tabular data, graph data incorporates both individual node attributes and pertinent structural information, offering an efficient mechanism to represent and analyze complex interrelationships among individuals [61]. Consequently, recent years have witnessed a surge of interest in the development and application of graph algorithms specifically designed for graph data. Among them, graph neural networks (GNNs) have shown great ability in modeling graph-structural data [16, 50], consistently delivering exceptional performance across a diverse range of tasks and applications [41]. Nevertheless, like many ML methodologies, GNNs have been observed to potentially discriminate against certain populations as identified by the *sensitive attribute* (*e.g.,* gender or race), leading to substantial ethical considerations.

To mitigate discrimination in GNNs, existing works primarily leverage statistical fairness notions to address bias in graph representation learning [42, 6, 48]. Their foundation lies in the assumption that bias originates solely from sensitive attributes, aiming to achieve predictions that are statistically equitable across subgroups. However, this strategy largely overlooks the widespread existence of labeling bias, where the labels of the samples are affected by factors unrelated to their determination, such as statistical anomalies [26]. Recent works [53, 46] have thus extended the concept of counterfactual fairness [17] to graphs, seeking to overcome the limitation in the presence of labeling bias by considering the causal relationships between variables. Specifically, this adaptation aims to ensure that nodes and their corresponding counterfactual instances (different versions of the nodes) receive consistent prediction results [48]. For example, in a job recruitment scenario, two candidates with different sensitive attribute values but similar qualifications should have equal hiring opportunities.

The existing counterfactual generating works predominantly generate instances by directly flipping sensitive attributes or perturbing node features. For instance, NIFTY [1] introduces perturbations to sensitive attributes to maximize the similarity between original and altered representations, thereby promoting invariance. Similarly, GEAR [22] employs GraphVAE [43] to minimize the discrepancy between original and counterfactual representations to eliminate the impact of sensitive attributes. Despite these advancements, these methods often produce potentially unauthentic counterfactuals [9, 5]. This is attributed to the fact that counterfactual inference is essentially an unsupervised learning task, and these methods tend to rely on oversimplified causal models that neglect unobserved hidden confounders, which

[*] Corresponding author
Email: {ziwang, wenbin.zhang}@fiu.edu

affect both the historical choice of treatment and the outcomes, thus preventing the accurate inference of causal effects [40]. For instance, socio-economic status, although unobserved, can influence both the type of medication a patient has access to and the patient's overall health. Without accounting for socio-economic status, it is challenging to isolate the causal effect of medications on health outcomes. Consequently, during counterfactual fairness assessment, it becomes problematic to discern whether a change in a patient's healthcare decision is due to a modification in a sensitive attribute or a shift in a confounder.

Furthermore, existing works on graph counterfactual fairness often overlook the impact of graph structure bias in GNNs [38]. Typically, GNNs employ a uniform message-passing mechanism that aggregates information from neighboring nodes, thereby preserving the topology and node feature information [23]. However, this process can inadvertently amplify existing biases within the graph's structure. In particular, the message-passing approach tends to homogenize the representations of connected nodes. Consequently, nodes connected by intra-group edges, which often exhibit similar features, may become over-represented [7]. In contrast, nodes linked by inter-edges, typically characterized by differing attributes (*i.e.,* high-frequency signals), might be under-represented during this aggregation process [36]. This imbalance often leads to a diminished representation of nodes from diverse sensitivity groups in the final node embedding. Therefore, constructing node edges for counterfactual instances based on node feature similarity often results in nodes with the same sensitive attributes being more closely connected [29]. This practice can inadvertently lead to unintended inter-group isolation and introduce structural bias (*i.e.*, intra- and inter-group edge distribution drift).

To address these limitations, this paper explores the domain of graph counterfactual fairness, with a focus on the potential causal interactions between each sample and its neighboring nodes. In addition, the impact of a sample's hidden confounders on its counterfactual instances, along with the influence of graph structure bias on the generation of these instances, is specifically examined. This area is largely underexplored with unique challenges: **i) Complexity of counterfactual graph data:** Unlike tabular data, which typically follows the principle of being independent and identically distributed (I.I.D.), graph data encompasses both node information and structural interconnections. The intricate nature of these relationships among the nodes implies that creating counterfactual samples requires the generation of features in the corresponding counterfactual scenario but also its interconnections with other nodes. **ii) Identifying hidden confounders:** The key to accurately generating counterfactual instances lies in identifying hidden confounders. However, since hidden confounders are not observable, determining how to accurately identify the hidden confounders of a sample based on its observable attributes is crucial yet challenging for obtaining authentic counterfactual scenarios. **iii) Effective learning inter-group edges:** Disparities in the edge distribution of central nodes can lead to an overrepresentation of neighboring nodes with the same sensitive attributes as the central node during node aggregation. This, in turn, causes an over-correlation of node embeddings with sensitive attributes, posing significant challenges in effectively learning inter-group edges during node aggregation to avoid introducing topological bias.

In response to the aforementioned challenges, this paper proposes a novel framework for graph-based fair ML decision-making. *To the best of our knowledge, this is the first work to mitigate the multi-source bias arising from sensitive attributes, labeling processes, and graph structure in graph-based models by considering both hidden confounders and fair node aggregation.* Specifically, in addition to the causal relationship explored by existing methods, which takes into account the interplay between sensitive attributes, graph structure, and non-sensitive attributes, our proposed causal model further encompasses the presence of hidden confounders, as estimated from the observed node features and graph structure and corresponding influence on them. Subsequently, the graph structure, node features, and the identified hidden confounders are used to learn a counterfactual instance generation function. Additionally, to improve the structural realism of counterfactual instances, an adaptive subgraph extractor is introduced to extend the subgraph by including neighbors that are important to the target node, even if they are far away. Different filter channels with an adaptive encoder are also constructed to discriminately aggregate neighboring information along intra- and inter-edges, which avoids over-correlation of node representations with sensitive attributes, thereby enhancing the quality of node embedding. Finally, the generated counterfactual graphs are employed to ensure prediction consistency across real-world and counterfactual scenarios, achieving graph counterfactual fairness.

The **key contributions** of this paper are: i) We formulate a new graph counterfactual fairness problem that demands concurrent alleviation of algorithmic biases associated with sensitive attributes, labeling processes, and the inherent structure of the graph; ii) We introduce AGCG, a novel framework crafted to attain counterfactual fairness in graphs. By concurrently tackling biases through the identification of hidden confounders and the implementation of fair message passing, it provides a holistic approach to mitigating bias in graphs; and iii) We conduct extensive experiments on three real-world benchmark datasets and a synthetic dataset to demonstrate the superiority of our proposed framework in terms of both bias mitigation and node classification performance.

2. RELATED WORK

2.1 Graph Neural Networks

Graph neural networks have found widespread utility in various tasks involving graph-structured data, such as node classification [33, 16, 3], graph classification [32, 25], and link prediction [62]. Their superior performance is attributed to their ability to represent learning on graphs, with two primary approaches: spectral-based and spatial-based. Specifically, spectral-based approaches, grounded in graph theory, adapt convolution operations to graph data and rely on the graph Laplacian matrix or the adjacency matrix to capture structural information about the graph in the spectral domain [37, 36, 45]. For instance, Graph Convolutional Networks (GCN) simplify graph convolution using a first-order approximation of these operations [16]. In contrast, spatial-based GNNs, like GraphSage [10] and EGNN [19], focus on learning node representations by aggregating information from neighboring nodes. Despite the methodological differences, most GNNs involve message passing—a process of pattern extraction and interaction modeling within each layer. However, a major challenge for this framework is how to effectively aggregate node information from neighbors with different sensitive attributes. Existing uniform aggregation leads to suppression of information from neighboring nodes with different sensitive attributes, introducing structural biases that, in turn, affect the fairness and performance of downstream tasks.

2.2 Fairness on Graphs

Fairness in graphs has received intensive attention. Most of the existing methods are based on statistical fairness notations such as group fairness [39, 11, 47] and individual fairness [31, 28, 43]. Specifically, group fairness evaluates whether the outcome statistics of the classifiers are similar across different subgroups [34], while individual fairness ensures that similar individuals receive similar probability distributions over class labels [8]. Despite their great success, they inadequately address label bias. To this end, counterfactual fairness [17] leverages the causal theory to eliminate the root bias. Existing works on graph counterfactual fairness either generate counterfactual instances [58] or identify potential counterfactual instances within input dataset [48]. However, the former relies on oversimplified causal models that neglect unobservable hidden confounders, failing to capture authentic counterfactual scenarios [27]. Meanwhile, the latter incorrectly assumes the presence of authentic counterfactual instances within the input data, an assumption that may not hold true. Consequently, both approaches struggle to reflect authentic counterfactual scenarios.

To jointly address these challenges, we aim to generate authentic counterfactual instances by acknowledging the existence of hidden confounders, identifying them with a variational inference approach with Gaussian mixture priors, and incorporating the information when learning the generating functions of counterfactual instances. Furthermore, our approach addresses the root cause of graph structure bias, enhances fairness in the node aggregation process, and improves the quality of node embeddings with minimal impact on overall model performance.

3. NOTATION

Consider an undirected and unweighted input graph with n nodes as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges such that $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and $\mathbf{X}$ is the set of node features with $x_i \in \mathbb{R}^{1 \times D}$ representing the features of individual node i. Each node v_i has a binary sensitive attribute s_i, indicating whether node v_i belongs to a deprived group ($s_i = 0$) or favored group ($s_i = 1$), which is part of the feature set $\mathbf{X}$. We use $S \in \{0, 1\}^{N \times 1}$ ($S \in \mathbf{X}$) to denote the vector representing the sensitive attributes of nodes. The adjacency matrix of the graph $\mathcal{G}$ is denoted as $A \in \{0, 1\}^{n \times n}$, where $A_{i,j} = 1$ if there is an edge between nodes v_i and v_j, and 0 otherwise. An edge $A_{i,j}$ is classified as an intra-group edge if nodes v_i and v_j share the same sensitive attribute value, and as an inter-group edge otherwise. We let v_{syn}, and s_{syn} denote the generated node, its sensitive attribute, respectively. Additionally, let $C = [C_1, \ldots, C_n]$ denote the matrix of hidden confounders, where each $C_i \in \mathbb{R}^p$ represents the confounders for node v_i. Without loss of generality, we use $\mathcal{L} = \{v_1, v_2, \ldots, v_L\}$ to signify the set of $\mathcal{L}$ labeled vertices, accompanied by their observed labels $Y = \{y_1, \ldots, y_L\}$, with y_i denoting the ground-truth label of vertex v_i. We also use $\mathcal{U} = \{v_{L+1}, v_{L+2}, \ldots, v_{L+U}\}$ representing the set of $\mathcal{U}$ unlabeled vertices, and the predicted labels are $\hat{Y} = \{\hat{y}_1, \ldots, \hat{y}_L\}$. Also note that $\mathcal{L} \bigcup \mathcal{U} = \mathcal{V}$.

4. METHODOLOGY

4.1 Causal Model

Figure 2 depicts the causal model, which serves as the foundation of AGCG for fair counterfactual decision-making. *To the best of our knowledge, this is the first causal model that delves into the causal relationships between hidden confounders (C) and observable attributes: sensitive attributes (S), node features (X), graph structure (A), and ground-truth label (Y) in the realm of counterfactual fairness.* In the proposed model, each connection represents a deterministic causal link, indicating the direct influence of one variable on another. Through this framework, AGCG can discern potential modifications that would occur in the counterfactual world under different conditions. Below, we delineate the rationale and explanations that underpin the causal model.

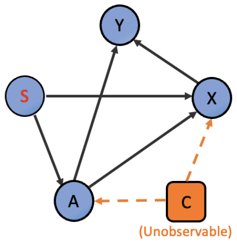


Figure 2: The causal model of AGCG.

- $A \leftarrow C \rightarrow X$: The hidden confounder C has implications for the graph structure A and node features X but does not directly affect sensitive attribute, nor can it be impacted by the graph structure and node features. For instance, a person's "bad temper" might influence his/her "blood pressure" and deteriorate his/her social relationships with others. However, it cannot change a person's "gender". Note that C represents unobservable features and might not always correspond to tangible entities in the real world. In addition, a causal path from C to ground-truth label Y is hypothesized to be mediated through observable variables (*i.e.*, A and X).

- $A \leftarrow S \rightarrow X$: Since the sensitive attribute S is typically determined at birth, there is no parent variable in the causal graph. Instead, S can only serve as the cause of other variables, which in turn influence the node's features X and graph structure A. For instance, the sensitive attribute "gender" cannot be caused by other features such as "height", whereas "gender" can influence "height". Similarly, on social networks, the "gender" of a person might skew their connections toward similarly gendered individuals, while these connections cannot change a person's "gender". Notably, S does not affect the hidden confounder C, *e.g.*, a person's "gender" does not affect their "health".

- $Y \leftarrow A \rightarrow X$: The graph structure A has implications for both node features X and ground-truth label Y, *i.e.*, potential for changes in one node to impact another. For

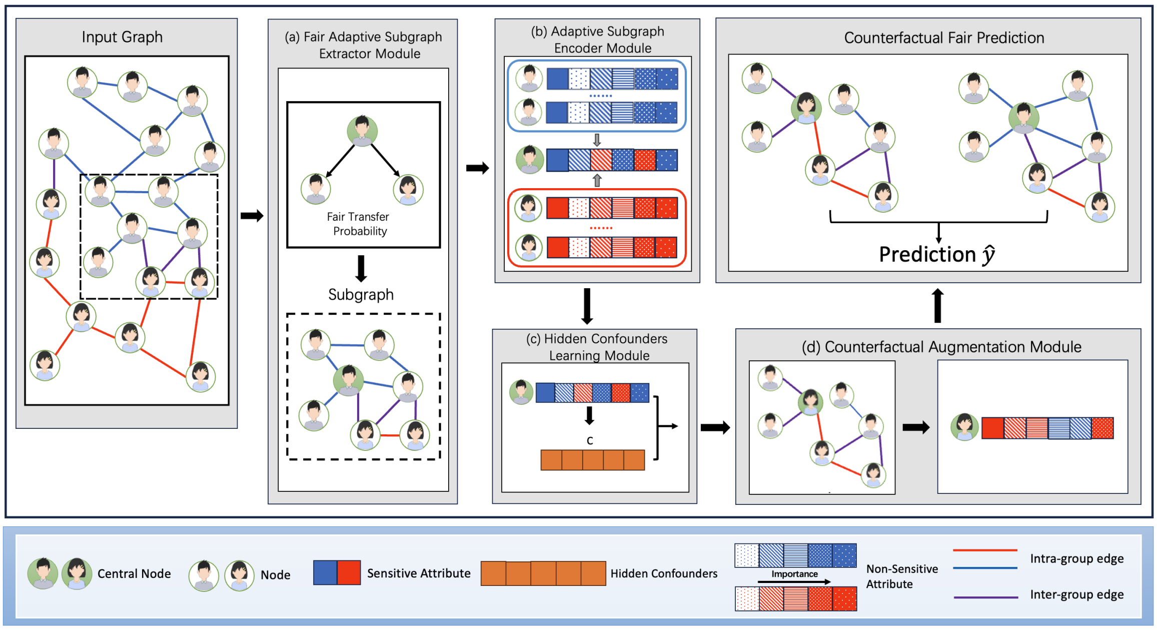

Input Graph
(a) Fair Adaptive Subgraph Extractor Module
Fair Transfer Probability
Subgraph
(b) Adaptive Subgraph Encoder Module
Counterfactual Fair Prediction
Prediction $\hat{y}$
(c) Hidden Confounders Learning Module
c
(d) Counterfactual Augmentation Module
Central Node
Node
Sensitive Attribute
Hidden Confounders
Importance
Non-Sensitive Attribute
Intra-group edge
Inter-group edge

Figure 1: Overview of the proposed AGCG framework. For each node, the framework first extracts its contextual subgraph. It then fair adaptively aggregates information from both intra- and inter-edges, infer the hidden confounders from the observed data and generates counterfactual instances using both the observed data and these hidden confounders.

example, if all of a person's friends have watched a particular movie, that person is also likely to watch it, thereby changing his/her movie-watching history.

- $X \rightarrow Y$: The node features X has impacts for the ground-truth label Y. Notably, a causal path from X to A would have similar total causal effects as a causal path from A to X. Therefore, for computational efficiency, we assume that there is no causal path from X to A.

4.2 AGCG: In a Nutshell

Expanding upon the established causal model, the proposed AGCG framework, comprising four modules, is illustrated in Figure 1: (a) The *Fair Adaptive Subgraph Extractor Module* (Section 4.3), which adaptively identifies contextual subgraphs relevant to each node; (b) The *Adaptive Subgraph Encoder Module* (Section 4.4), tailored to differentially aggregate information from intra-group and inter-group edges linked to each node; and (c) The *Hidden Confounders Learning Module* (Section 4.5), dedicated to inferring hidden confounders for each node based on its observed features and structure; (d) The *Counterfactual Augmentation Module* (Section 4.6), responsible for creating counterfactual instances by integrating observed graph data with the inferred hidden confounders. Subsequent sections will delve into the details of each module.

4.3 Fair Adaptive Subgraph Extractor Module

Learning causal models directly on large-scale graph data (*e.g.*, social networks) can be computationally expensive. To mitigate this complexity, a common strategy is to extract a local subgraph for each node, based on the assumption that a node is predominantly influenced by its immediate neighborhood [14]. Drawing upon previous work [39], the proposed subgraph extraction module operates under the premise that each node v_i depends minimally on nodes outside a certain "context subgraph". This context subgraph aims to retain essential structural and relational information relevant to v_i.

In addition, unlike approaches that restrict the neighborhood size (*e.g.*, to 1-hop neighbors), our method leverages importance scores ($ImpS$) to identify influential nodes, regardless of their distance. As depicted in Figure 1 (a), for a given central node (highlighted in green), we extract a context subgraph (shown within the dashed black rectangle) composed of the top-k most influential neighbor nodes based on $ImpS$. In doing so, we broaden the local neighborhood beyond immediate neighbors, incorporating distant yet informative nodes. This enhances both representation learning and counterfactual data augmentation.

To compute $ImpS$, we first construct a normalized adjacency matrix $\overline{A} = AD^{-1}$, where D is a diagonal degree matrix with entries $D_{i,i} = \sum_j A_{i,j}$. However, this standard normalization does not account for fairness concerns, as it fails to consider how nodes sharing certain sensitive attributes often form densely connected substructures. These substructures can distort transition probabilities, amplifying subgroup disparities within the extracted subgraphs [13]. To counteract this bias, a fairness constraint on the transition probabilities is enforced. Specifically, neighboring nodes are classified by their sensitive attributes, ensuring that the aggregate selection probabilities are balanced across these attributes, and this is formally imposed as:

$$\sum(P_{v_a}|\overline{A}_{a,j} = 1, s_a \in S_d) = \sum(P_{v_b}|\overline{A}_{b,j} = 1, s_b \in S_f) \quad (1)$$

where P_{v_a} and P_{v_b} are the transition probabilities to neighboring nodes in deprived (S_d) and favored (S_f) groups, respectively. With this fairness-aware normalization, $ImpS$ is then computed as:

$$ImpS = \alpha(I - (1-\alpha)\overline{A})^{-1} \tag{2}$$

where I is the identity matrix and $\alpha \in [0, 1]$ controls the restart probability from the central node. Each entry $ImpS_{i,j}$ measures the importance of node v_j to node v_i, and $ImpS_{i,:}$ denotes the importance vector for node v_i. This computation is performed as a pre-processing step, thus not incurring additional overhead during model training. Armed with these importance scores, our Adaptive Subgraph Extraction module selects the top-k high-$ImpS$ nodes for each central node v_i to form its context subgraph $\mathcal{G}_{v_i}$.

4.4 Adaptive Subgraph Encoder Module

After extracting the subgraphs, AGCG aggregates information from each central node's neighbors to obtain final node embeddings. Existing approaches [4, 21] often apply uniform message-passing over both intra- and inter-group edges without differentiating among distinct information frequencies (*e.g.*, low-frequency, high-frequency, and identity information). This uniform treatment may cause the learning process to be dominated by the majority edge type (*i.e.*, intra-group edges) while neglecting critical signals from less-represented edges (*i.e.*, inter-group edges).

To address this issue, we propose a new subgraph encoder that considers the disproportion in intra- and inter-group edge distributions. By distinctly handling information derived from these edges, our encoder reduces bias and ensures that embeddings adequately represent diverse subgroups. For example, as shown in Figure 1(b), the encoder separately aggregates information from intra-group edges (in the blue box) and inter-group edges (in the red box). This approach enriches the final embedding for the central node with valuable signals from nodes having different sensitive attributes, mitigating bias introduced by skewed edge distributions. Specifically, we introduce three types of learnable weights to handle different frequency components: i) Low-frequency weight (ω_L). We capture commonalities between the central node and its neighbors by concatenating their transformed features. Mathematically denoted as $\omega_L(v_i, k_i)$ $= \sigma(\mathbf{u}_L^\top(W_L z_L(v_i, k_i)))$, where $\mathbf{u}_L$ is a learnable vector. ii) High-frequency weight (ω_H). To highlight differences between neighbors, especially those with distinct sensitive attributes, we incorporate a negative sign on the neighbor's features before transformation. Mathematically, it represent as $\omega_H(v_i, k_i) = \sigma\big(\mathbf{u}_H^\top(W_H(-h_{k_i}^{(l-1)}))\big)$. iii) Identity weight ($\omega_I$). To preserve the central node's inherent characteristics, we define: $\omega_I(v_i, k_i) = \sigma(\mathbf{u}_I^\top(W_I h_{v_i}^{(l-1)}))$. Here, $\sigma(\cdot)$ is the sigmoid activation function, W_L, W_H, W_I are layer-specific transformation matrices for different frequency components, and $\mathbf{u}_L, \mathbf{u}_H, \mathbf{u}_I$ are learnable parameter vectors.

To effectively integrate these weights, we normalize them across the three information types for each node pair (v_i, k_i): $\hat{\omega}_{(v_i,k_i)} = \big[\varpi_L(v_i, k_i), \varpi_H(v_i, k_i), \varpi_I(v_i, k_i)\big]$, where $\varpi_{a\in\{L,H,I\},(v_i,k_i)}$ is calculated as: $\varpi_a(v_i, k_i) =$ $\text{softmax}\,(\omega_a(v_i, k_i))$. The obtained weighting vector $\hat{\omega}_{(v_i,k_i)}$ is used to aggregate the multi-frequency information from neighboring nodes to compute the central node embedding:

$$\begin{aligned} h^l_{v_i} =& \text{UPD}^l_k\big(\pi h^{l-1}_{v_i}, \text{AGG}_{k_j\in\mathcal{G}_{v_i}} \\ &\big(\hat{\omega}^{(l)}_{(v_i,k_j)}\, \text{ReLU}\,\big(W_R[W_L h^{l-1}_{k_j}, W_H h^{l-1}_{k_j} W_I h^{l-1}_{k_j}]\big)\big)\big) \end{aligned} \tag{3}$$

where π is a hyperparameter, $W_R \in \mathbb{R}^{d_l\times 3d_l}$ denotes the projected matrix, integrating the embeddings from layer $l-1$, and $h^l_{v_i}$ denotes the aggregated neighboring embedding of node $k_i \in \mathcal{G}_{v_i}$ after superimposing the l layer encoder.

To effectively train a fair adaptive subgraph encoder module, we train it with an adjacency matrix reconstruction task. Considering the sparsity of positive edges (*e.g.,* existing edges), we also adopt negative sampling (*e.g.*, non-existing edges) to train our module. To ensure the number of positive samples ($\mathcal{E}^+$) is the same as negative samples ($\mathcal{E}^-$), we randomly choose $|\mathcal{E}^+|$ negative edges from the total negative edges as negative samples. The reconstruction loss $\mathcal{L}_{rec}$ is calculated as follows:

$$\mathcal{L}_{rec} = \frac{1}{|\mathcal{E}^+| + |\mathcal{E}^-|} \sum_{e_{ij}\in\mathcal{E}} L(e_{ij}, \hat{e}_{ij}) \tag{4}$$

where $\hat{e}_{ij}$ and e_{ij} are the predicted and observed edge of input graph $\mathcal{G}$, respectively. This approach effectively trains the model to distinguish between existing and non-existing links, enhancing its ability to accurately reconstruct the graph structure.

4.5 Hidden Confounders Learning Module

In this section, we discuss how AGCG generates authentic counterfactual instances to achieve graph counterfactual fairness. According to the causal analysis in Section 4.1, generating these instances relies on accurately approximating the joint distribution $P(C, A, X, S) =$ $P(C|X, A, S)P(A, X, S)$. However, this task is complicated by the unobservability of hidden confounders C and the computational infeasibility of directly calculating the marginal likelihood $P(X, A, S)$ due to the need to integrate C. To this end, we optimize the Evidence Lower Bound [15] (ELBO) related to the marginal log-likelihood of the observable graph data (*e.g.*, A, X, S), which allows us to effectively approximate and recover the joint distribution $P(C, A, X, S)$, thus ensuring the authenticity of the counterfactual instances generated, as demonstrated in Equation 5:

$$\log P(A, X|S) \geq \mathbb{E}_{Q(C|A,X,S)}[\log P(C, A, X, S)] - \mathbb{E}_{Q(C|A,X,S)}[\log Q(C|A, X, S)] \tag{5}$$

where $Q(C|A, X, S)$ denotes the variational distribution that uses a parametric family of distributions to approximate the intractable posterior distribution $P(C|A, X, S)$. This strategy allows us to sample C from its posterior given the observable variables (*e.g.*, A, X, S), as formalized in Equation 6:

$$P(C|X, A, S) = \frac{P(X, A, S|C)P(C)}{P(X, A, S)} \tag{6}$$

Building upon this approximation framework, we model the joint distribution $P(C, A, X, S)$ consistently with our proposed causal model (as shown in Figure 2), as outlined in Equation 7:

$$P(C, A, X, S) = P(C)P(S)P(A|C, S)P(X|A, C, S) \tag{7}$$

where $P(A|C, S)$ and $P(X|A, C, S)$ are the graph structure generation function G_A and the node features generation function G_X, respectively, to be detailed in Section 4.6. In addition, the generative and inference model parameters are learned simultaneously by maximizing the ELBO.

However, this modeling strategy relies on the assumption that the VAE model is identifiable, a premise that has not been fully established [20, 24]. This is attributed to the fact that multiple parameter sets can yield models with identical marginal data and prior distributions, yet differ significantly in the hidden confounder C. Consequently, obtaining the true joint distribution $P(C, A, X, S)$ only using VAE is not feasible. To ensure the model in Equation 7 is identifiable, we specify a Gaussian mixture prior to the hidden confounder c_i associated with v_i. To achieve identifiability and capture more complex latent patterns, we impose a mixture of Gaussians prior on the hidden confounder c_i. Specifically, a discrete random variable selects one among a finite set of candidate Gaussian components, each characterized by its own mean vector and covariance matrix. The relative likelihood of choosing each component is governed by a set of mixing proportions that sum to one. By marginalizing over these discrete assignments, the effective prior emerges as a weighted combination of multiple Gaussian densities. By employing this mixture prior, the latent space can accommodate multiple modes and subtle variations in the underlying data, helping to alleviate identifiability issues. This setup thus not only supports a richer and more nuanced characterization of the hidden confounders but also provides a structured probabilistic foundation for improved variational inference. Building on this structured prior, we define a categorical distribution to describe the allocation of weights among the Gaussian components, facilitating the necessary probabilistic framework for effective variational inference, where probabilities of the individual components are formulated using a categorical distribution. Moreover, let T be the vector containing the component values for all nodes within the graph, and the variational distribution $Q(C, T|A, X)$ can be factorized as:

$$Q(C, T|A, X, S) = Q(C|A, X, S)Q(T|A, X, S) \tag{8}$$

Building on this, the updated ELBO of our framework can be formally described as:

$$\begin{aligned}\log P(A, X|S) &= \log \int\int P(C)P(S)P(A|C, S)P(X|A, C, S)\, dC\, dT \\ &\geq \mathbb{E}_{Q(C,T|A,X,S)}\left[\log \frac{P(A, X, C, T|S)}{Q(C, T|A, X, S)}\right]\end{aligned} \tag{9}$$

where the values of $\log P(A, X|S)$ correlate positively with the reality of both the graph structure and node features, while $\mathbb{E}_{Q(C,T|A,X,S)}$ denotes the expectation with respect to the variational distribution Q. Furthermore, $P(A, X, C, T|S)$ represents the joint distribution between the observed data, while $P(C, T|A, X, S)$ denotes the posterior distribution of the hidden confounders.

Using the factorization of the variational distribution, the updated ELBO of our framework can be formally described as:

$$\log P(A, X|S) \geq \mathbb{E}_{Q(C,T|A,X,S)}\left[\log \frac{P(T)P(C|T)P(A|C, S)P(X|A, C, S)}{Q(C, T|A, X, S)}\right] \tag{10}$$

4.6 Counterfactual Augmentation Module

With hidden confounders identified, AGCG proceeds to generate counterfactual instance for each node v_i, which is subsequently used to train downstream classifiers to achieve graph counterfactual fairness. As shown in Figure 1 (d), two generating functions are utilized to generate the graph structure A (left), and node features X (right) of the counterfactual instance, respectively. Specifically, given an observed graph $\mathcal{G} = \{A, X, S\}$, the sensitive attributes S' of each node v_i are flipped (*i.e.*, $S'_i = \neg S_i$), then utilized along with the obtained hidden confounder C to generate the adjacency matrix A' according to $G_A(C, S')$, representing the topology of the counterfactual subgraph (the ego graph of the counterfactual counterpart of v_i) as follows:

$$A'(v_{syn}, v_j) = \frac{1}{1 + e^{-C_{v_{syn}} \cdot C_{v_j}^T}} \tag{11}$$

where $C_{v_{syn}}$ and C_{v_j} represent the hidden confounder of a generated counterfactual instance and of other nodes within the observed graph $\mathcal{G}$, respectively. In addition, to enhance genuineness, real world graph topology distributions are incorporated. Specifically, nodes sharing the same sensitive attribute value (*i.e.*, within the same subgroup) are more likely to form connections. To reflect this real-world phenomenon, two thresholds are established: η_1 for nodes with the same sensitive attribute value and η_2 for nodes with different sensitive attribute values. These thresholds represent the probability that a connection exists between pairs of nodes, depending on whether they share sensitive attribute value, with the exact values to be fine-tuned according to the dataset's distribution. This strategy, informed by the observed input graph topology, aims to ensure that the synthesized graph structures reflect realistic connectivity patterns and avoid overly sparse connections between nodes with differing sensitive attributes, thereby improving the realism and fairness of the counterfactual graph structure.

In terms of generating node features $X' = G_X(C, A', S')$, the process is based on:

$$\begin{aligned} X'^{(q)}_i = & G_{X1}(c_i, s_i, X'^{(q-1)}_i) + \qquad (12) \\ & AGG_{v_j \in \mathcal{G}_i^{(q)}}(G_{X2}(c_j, s_j, X'^{(q-1)}_j)) \qquad (13) \end{aligned}$$

where $X'^{(q-1)}_i$ denote the $(q-1)^{th}$ layer feature of node v_i, and $AGG(\cdot)$ denotes an aggregation function that maps the information from all neighboring nodes to a single vector. In addition, G_{X1} and G_{X2} are two piecewise affine transformation functions, *e.g.,* multilayer perceptrons with leaky $ReLU$ activations.

Building on these generated authentic counterfactual instances, our classifier is then trained with real samples and their counterfactual counterparts to ensure consistent predicted labels for both. The corresponding loss function $\mathcal{L}_{fair}$ is denoted as:

$$\mathcal{L}_{fair} = -\frac{1}{N}\sum_{i=1}^{N}\left[\hat{y}_i \log(\hat{y}'_i) + (1-\hat{y}_i)\log(1-\hat{y}'_i)\right] \quad (14)$$

where $\hat{y}_i$ is the predicted label of the real sample, and $\hat{y}'_i$ is for the counterfactual sample. The fair loss function minimizes the prediction discrepancy between the predictions for real and counterfactual instances, thereby promoting fairness.

To maintain the utility of AGCG, we adopt the cross-entropy loss as the utility loss $\mathcal{L}_{utility}$, defined as:

$$\mathcal{L}_{utility} = \frac{1}{|\mathcal{V}_L|}\sum_{v_i \in \mathcal{V}_L} -(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)) \quad (15)$$

The overall objective function is ultimately derived from all the loss functions described above.

5. EXPERIMENT

5.1 Datasets

Experiments are conducted on three real-world datasets and a synthetic dataset. For the real-world datasets: i) The **German** dataset [2] contains credit information of clients at a German bank. Each node represents a client, and each edge denotes the similarity between two clients' credit accounts. The sensitive attribute is the clients' gender, aiming to classify clients into good versus bad credit risks. ii) The **Credit** dataset [52] comprises individuals' default payment information, where each node signifies an individual, and edges denotes the similarity in their expenditure and payment patterns. The sensitive attribute is age, with the objective of predicting whether an individual's default mode of payment is via credit card. iii) The **Bail** dataset [1] presents data related to defendants granted bail in U.S. state courts. Each node corresponds to a defendant, and an edge connecting two nodes signifies similarities in their criminal records and demographic details. The race of the defendants is used as the sensitive attribute, with the goal of classifying defendants into two categories: those suitable for bail and those who are not.

As real-world datasets lack ground-truth counterfactuals, a synthetic dataset is constructed using the proposed causal model. This gives us accurate counterfactuals for each node, enabling precise assessment of generated instances. In our setup, we consider the binary sensitive attributes and labels, which are generated based on a Bernoulli distribution, *i.e.*, $s_i \sim Bernoulli(p_s)$. Next, we begin by drawing latent factors C from a Gaussian mixture model with Z elements. For each node, we use G_X and G_A to generate node feature X and adjacency matrix A. Node labels are generated in the same way as node features. This way allows us to manipulate various parameters, including the sensitive attribute probability, label probability, and feature dimensions. Table 1 provides detailed statistics of these four datasets.

Table 1: Summary of the datasets used in the experiments.

Dataset	German	Credit	Bail	Synthetic
Vertices	1,000	30,000	18,876	2,000
Edges	21,742	137,377	311,870	4,570
Feature dimension	27	13	18	25
Average Degree	44.5	10	34	4.9
Sensitive Attribute	Gender	Age	Race	Gender

5.2 Baselines

To assess AGCG, we compare it against eight state-of-the-art node classification methods, categorized into three groups: i) Vanilla graph model: **GCN** [16], **GraphSAGE** [10], and **GIN** [51]. ii) Fair Node Classification Methods: **Graphair** [12] aims to use adversarial learning to automatically produce fair graph data that can trick the discriminator. **FairAGG** [63] implements a fair aggregation scheme based on the Shapley value to ensure group fairness. iii) Graph Counterfactual Fairness Methods:**NIFTY** [1] create counterfactuals by introducing perturbations to sensitive attributes, thereby enhancing model fairness. **RFCGNN** [44] learns a fair node representation by identifying counterfactual instances and sensitive attribute-related information masking, and **FDGNN** [39] utilizes counterfactual samples to learn disentangled node representation to mitigate the multi-source biases.

5.3 Evaluation Metrics

Both fairness and predictive performance are evaluated, with Statistical Parity Differences (SPD) [18] and Equal Opportunity Differences (EOD) [11], with values close to zero indicating better fairness. We utilize AUC and F1-score to measure the performance on node classification tasks, with higher values indicating better performance.

5.4 Experiment results

Comparison Study. Table 2 presents the performance of node classification and fairness, including the standard deviation from 10 experiments, along with the average results. As can be seen, AGCG is affirmed by the empirical results as highly effective. Specifically, AGCG consistently achieves top rankings for fairness metrics across all datasets, when compared with other baseline methods. From the perspective of model utility, AGCG demonstrates comparable results in the F1-score, while the AUC is higher than some baselines. AGCG's advantage stems from its accurate causal model, which accounts for hidden confounders, leading to the generation of authentic counterfactual scenarios that improve the model's graph counterfactual fairness. Furthermore, AGCG effectively mitigates graph structure bias, reducing the likelihood of node embeddings being overly influenced by sensitive attributes, and efficiently leverages key information from neighbors with differing sensitive attributes. Overall, AGCG exhibits good performance in balancing the trade-off between prediction accuracy and fairness.

Ablation Study. To evaluate the effectiveness of the individual components in AGCG, an ablation study was conducted. Initially, the significance of the adaptive subgraph encoder model was examined. For comparison, this component was removed and replaced with the AGCG-NAE variant, utilizing a standard encoder, such as performing uni-

Table 2: Predictive and fairness performance for AGCG and baselines across real-world datasets and synthetic datasets.

Dataset	Methods / Metrics	Vanilla Methods			Fair Node Classification Methods		Graph Counterfactual Fairness Methods			
		GCN	GraphSAGE	GIN	Graphair	FairAGG	NIFTY	RFCGNN	FDGNN	**AGCG**
German	**AUC** (↑)	0.654 ± 0.015	**0.781** ± 0.008	0.734 ± 0.012	0.718± 0.054	0.704 ± 0.020	0.736 ± 0.041	0.747 ± 0.029	**0.781** ± 0.022	0.744 ± 0.048
	F1-Score (↑)	0.786 ± 0.012	0.817 ± 0.019	0.812 ± 0.015	0.813 ± 0.012	0.781 ± 0.014	0.792 ± 0.019	0.823 ± 0.012	**0.837** ± 0.021	0.828 ± 0.037
	SPD(↓)	0.364 ± 0.052	0.231 ± 0.058	0.148 ± 0.046	0.084 ± 0.073	0.063 ± 0.047	0.077 ± 0.028	0.067 ± 0.017	0.058 ± 0.010	**0.057** ± 0.010
	EOD(↓)	0.312 ± 0.041	0.157 ± 0.056	0.091 ± 0.037	0.058 ± 0.023	0.036 ± 0.038	0.049 ± 0.023	0.041 ± 0.016	0.024 ± 0.009	**0.021** ± 0.017
Credit	**AUC** (↑)	0.707 ± 0.017	**0.767** ± 0.013	0.728 ± 0.013	0.758 ± 0.047	0.721 ± 0.022	0.727 ± 0.024	0.743 ± 0.033	0.747 ± 0.031	0.734 ± 0.022
	F1-Score (↑)	0.835 ± 0.028	0.859 ± 0.011	0.809 ± 0.018	0.728 ± 0.072	0.747 ± 0.042	0.806 ± 0.012	0.849 ± 0.049	**0.861** ± 0.048	0.859 ± 0.031
	SPD(↓)	0.108 ± 0.035	0.113 ± 0.037	0.132 ± 0.037	0.085 ± 0.034	0.074 ± 0.036	0.094 ± 0.017	0.074 ± 0.047	**0.056** ± 0.024	0.063 ± 0.024
	EOD(↓)	0.096 ± 0.035	0.124 ± 0.047	0.128 ± 0.047	0.088 ± 0.035	0.056 ± 0.021	0.113 ± 0.027	0.064 ± 0.016	0.047 ± 0.016	**0.043** ± 0.013
Bail	**AUC** (↑)	0.871 ± 0.019	0.894 ± 0.021	0.768 ± 0.067	0.822 ± 0.023	0.803 ± 0.016	0.796 ± 0.008	**0.896** ± 0.017	0.894 ± 0.013	0.866 ± 0.024
	F1-Score (↑)	0.784 ± 0.022	0.793 ± 0.031	0.658 ± 0.088	0.763 ± 0.038	0.743 ± 0.024	0.674 ± 0.062	**0.802** ± 0.032	0.785 ± 0.022	0.768 ± 0.057
	SPD(↓)	0.093± 0.015	0.086 ± 0.035	0.072 ± 0.037	0.051 ± 0.033	0.047 ± 0.035	0.035 ± 0.037	0.031 ± 0.013	0.025 ± 0.011	**0.023** ± 0.018
	EOD(↓)	0.044 ± 0.015	0.041 ± 0.022	0.043 ± 0.027	0.045 ± 0.033	0.036 ± 0.024	0.028 ± 0.023	0.024 ± 0.016	0.020 ± 0.014	**0.018** ± 0.008
Synthetic	**AUC** (↑)	0.653 ± 0.013	**0.705** ± 0.017	0.693 ± 0.015	0.662 ± 0.029	0.657 ± 0.024	0.702 ± 0.041	0.663 ± 0.030	0.695 ± 0.037	0.703 ± 0.033
	F1-Score (↑)	0.657 ± 0.024	0.685 ± 0.019	0.673 ± 0.024	0.676 ± 0.028	0.631 ± 0.032	0.713 ± 0.042	0.689 ± 0.032	0.724 ± 0.047	**0.727** ± 0.050
	SPD(↓)	0.146 ± 0.035	0.138 ± 0.042	0.248 ± 0.055	0.054 ± 0.023	0.040 ± 0.027	0.045 ± 0.024	0.061 ± 0.031	0.041 ± 0.021	**0.032** ± 0.011
	EOD(↓)	0.128 ± 0.032	0.114 ± 0.027	0.183 ± 0.048	0.022 ± 0.033	0.028 ± 0.037	0.038 ± 0.011	0.031 ± 0.017	0.023 ± 0.019	**0.018** ± 0.012

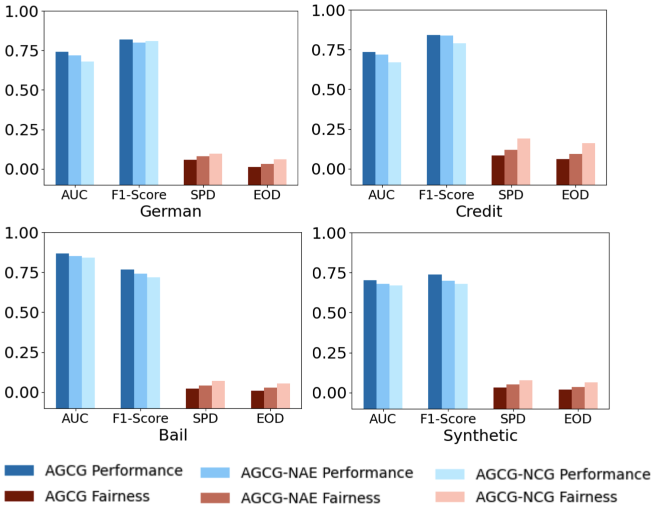


Figure 4: Ablation study results for AGCG, AGCG-NAE, and AGCG-NCG.

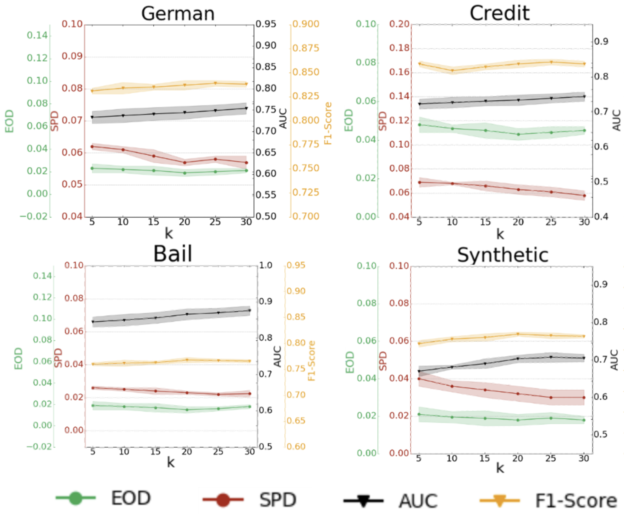


Figure 5: Parameter study on the choice of k-value.

form messaging for both intra- and inter-group edges. As depicted in Figure 4, the fairness of AGCG-NAE notably declined. This decrease is attributed to the model's inability to adequately learn information from neighboring nodes with differing sensitive attributes during the aggregation process, thus introducing structural bias. Additionally, AGCG-NAE can result in the over association of the node representations with sensitive attributes, degradation of the quality of the generated counterfactual instances, and, consequently, model performance. Subsequently, the importance of the counterfactual augmentation model was evaluated by creating an AGCG-NCG variant in its absence. As shown in Figure 4, AGCG-NCG also exhibited a significant drop in fairness performance, underscoring the critical role of the fairness module in mitigating potential biases.

Effect of Different k Values. In the experiments, we evaluated the impact of varying subgraph sizes k, in $\{5, 10, 15, 20, 25, 30\}$, while keeping all other training factors constant. The classification and fairness performance on all datasets are depicted in Figure 5. It is observed that the model achieves better fairness with relatively larger subgraph sizes. Specifically, the model exhibits more substantial fairness enhancements as the subgraph size increases up to 20. However, this improvement becomes less significant when the subgraph size exceeds 20. This is because nodes with $ImpS$ scores ranked after the top 20 hold limited importance to the center node. Consequently, expanding the subgraph beyond 20 nodes yields negligible gains in classification and fairness performance.

6. CONCLUSION

This paper introduces AGCG, a novel graph counterfactual fairness framework designed to enhance the fairness of GNNs. AGCG addresses a critical gap in existing graph counterfactual fairness works, *i.e.,* oversimplified causal models that overlook hidden confounders. Furthermore, by explicitly considering and mitigating the effects of graph structural biases, AGCG ensures consistent representation of different subgroups in node embeddings. The AGCG framework achieves graph counterfactual fairness by simultaneously learning from the original sample and its corresponding authentic counterfactual sample. Experimental evaluations, performed on both synthetic and real-world graph data, substantiate the efficacy of our proposed method in maintaining superior prediction performance while enhancing fairness. This work provides a new perspective on achieving counterfactual fairness in graph data, contributing to the ongoing development of fair GNNs.

Acknowledgement

This work was supported in part by the National Science Foundation (NSF) under Grant No. 2245895.

7. REFERENCES

[1] Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. "Towards a unified framework for fair and stable graph representation learning". In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 2114–2124.

[2] Arthur Asuncion and David Newman. *UCI machine learning repository*. 2007.

[3] Smriti Bhagat, Graham Cormode, and S Muthukrishnan. "Node classification in social networks". In: *arXiv preprint arXiv:1101.3291* (2011).

[4] Deyu Bo et al. "Beyond low-frequency information in graph convolutional networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 5. 2021, pp. 3950–3957.

[5] Sribala Vidyadhari Chinta et al. "FairAIED: Navigating fairness, bias, and ethics in educational AI applications". In: *arXiv preprint arXiv:2407.18745* (2024).

[6] Zhibo Chu, Zichong Wang, and Wenbin Zhang. "Fairness in Large Language Models: A Taxonomic Survey". In: *ACM SIGKDD Explorations Newsletter, 2024* (2024), pp. 34–48.

[7] Enyan Dai and Suhang Wang. "Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information". In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 2021, pp. 680–688.

[8] Cynthia Dwork et al. "Fairness through awareness". In: *Proceedings of the 3rd innovations in theoretical computer science conference*. 2012, pp. 214–226.

[9] Sander Greenland, Judea Pearl, and James M Robins. "Confounding and collapsibility in causal inference". In: *Statistical science* 14.1 (1999), pp. 29–46.

[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs". In: *Advances in neural information processing systems* 30 (2017).

[11] Moritz Hardt, Eric Price, and Nati Srebro. "Equality of opportunity in supervised learning". In: *Advances in neural information processing systems* 29 (2016).

[12] Fenyu Hu et al. "Graphair: Graph representation learning with neighborhood aggregation and interaction". In: *Pattern Recognition* 112 (2021), p. 107745.

[13] Zhimeng Jiang et al. "Fmp: Toward fair graph message passing against topology bias". In: *arXiv preprint arXiv:2202.04187* (2022).

[14] Yizhu Jiao et al. "Sub-graph contrast for scalable self-supervised graph representation learning". In: *2020 IEEE international conference on data mining (ICDM)*. IEEE. 2020, pp. 222–231.

[15] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[16] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[17] Matt J Kusner et al. "Counterfactual fairness". In: *Advances in neural information processing systems* 30 (2017).

[18] Tai Le Quy et al. "A survey on datasets for fairness-aware machine learning". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 12.3 (2022), e1452.

[19] Yuan Li et al. "EGNN: Constructing explainable graph neural networks via knowledge distillation". In: *Knowledge-Based Systems* 241 (2022), p. 108345.

[20] Christos Louizos et al. "Causal effect inference with deep latent-variable models". In: *Advances in neural information processing systems* 30 (2017).

[21] Sitao Luan et al. "Revisiting heterophily for graph neural networks". In: *Advances in neural information processing systems* 35 (2022), pp. 1362–1375.

[22] Jing Ma et al. "Learning fair node representations with graph counterfactual fairness". In: *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 2022, pp. 695–703.

[23] Yao Ma et al. "A unified view on graph neural networks as graph signal denoising". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2021, pp. 1202–1211.

[24] David Madras et al. "Fairness through causal awareness: Learning causal latent-variable models for biased data". In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pp. 349–358.

[25] Christopher Morris et al. "Weisfeiler and leman go neural: Higher-order graph neural networks". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 4602–4609.

[26] Alexandra Olteanu et al. "Social data: Biases, methodological pitfalls, and ethical boundaries". In: *Frontiers in big data* 2 (2019), p. 13.

[27] Judea Pearl. "Simpson's paradox, confounding, and collapibility". In: *Causality: models, reasoning and inference* (2009), pp. 173–200.

[28] Felix Petersen et al. "Post-processing for individual fairness". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 25944–25955.

[29] Tahleen Rahman et al. "Fairwalk: Towards fair graph embedding". In: (2019).

[30] Nripsuta Ani Saxena, Wenbin Zhang, and Cyrus Shahabi. "Missed opportunities in fair AI". In: *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM. 2023, pp. 961–964.

[31] Saeed Sharifi-Malvajerdi, Michael Kearns, and Aaron Roth. "Average individual fairness: Algorithms, generalization and experiments". In: *Advances in neural information processing systems* 32 (2019).

[32] Yongduo Sui et al. "Causal attention for interpretable and generalizable graph classification". In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, pp. 1696–1705.

[33] Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[34] S Verma and J Rubin. "Fairness Definitions Explained. 2018 IEEE". In: *ACM International Workshop on Software Fairness (FairWare), Gothenburg, Sweden.* 2018.

[35] Huaiyu Wan et al. "Aminer: Search and mining of academic social networks". In: *Data Intelligence* 1.1 (2019), pp. 58–76.

[36] Tao Wang et al. "Powerful graph convolutional networks with adaptive propagation mechanism for homophily and heterophily". In: *Proceedings of the AAAI conference on artificial intelligence.* Vol. 36. 4. 2022, pp. 4210–4218.

[37] Zichong Wang and Wenbin Zhang. "Group Fairness with Individual and Censorship Constraints". In: *27th European Conference on Artificial Intelligence.* 2024.

[38] Zichong Wang et al. ": Fairness-Aware Graph Generative Adversarial Networks". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer. 2023, pp. 259–275.

[39] Zichong Wang et al. "Advancing graph counterfactual fairness through fair representation learning". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer. 2024, pp. 40–58.

[40] Zichong Wang et al. "FG-SMOTE: Towards Fair Node Classification with Graph Neural Network". In: *ACM SIGKDD Explorations Newsletter, 2025* (2025).

[41] Zichong Wang et al. "FG^{2}AN: Fairness-aware Graph Generative Adversarial Networks". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD).* Turin, Italy, 2023.

[42] Zichong Wang et al. "History, Development, and Principles of Large Language Models-An Introductory Survey". In: *AI and Ethics, 2024* (2024).

[43] Zichong Wang et al. "Individual Fairness with Group Awareness Under Uncertainty". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer Nature Switzerland. 2024, pp. 89–106.

[44] Zichong Wang et al. "Mitigating multisource biases in graph neural networks via real counterfactual samples". In: *2023 IEEE International Conference on Data Mining (ICDM).* IEEE. 2023, pp. 638–647.

[45] Zichong Wang et al. "Preventing Discriminatory Decision-making in Evolving Data Streams". In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT).* 2023.

[46] Zichong Wang et al. "Toward Fair Graph Neural Networks via Real Counterfactual Samples". In: *Knowledge and Information Systems* (2024), pp. 1–25.

[47] Zichong Wang et al. "Towards Fair Graph Pooling with Group and Individual Awareness". In: *proceedings of the AAAI conference on artificial intelligence.* 2024.

[48] Zichong Wang et al. "Towards fair machine learning software: Understanding and addressing model bias through counterfactual thinking". In: *arXiv preprint arXiv:2302.08018* (2023).

[49] Jiancan Wu et al. "Self-supervised graph learning for recommendation". In: *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval.* 2021, pp. 726–735.

[50] Zonghan Wu et al. "A comprehensive survey on graph neural networks". In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24.

[51] Keyulu Xu et al. "How powerful are graph neural networks?" In: *arXiv preprint arXiv:1810.00826* (2018).

[52] I-Cheng Yeh and Che-hui Lien. "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients". In: *Expert systems with applications* 36.2 (2009), pp. 2473–2480.

[53] Zhipeng Yin, Zichong Wang, and Wenbin Zhang. "Improving Fairness in Machine Learning Software via Counterfactual Fairness Thinking". In: *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings.* 2024, pp. 420–421.

[54] Si Zhang et al. "Hidden: hierarchical dense subgraph detection with application to financial fraud detection". In: *Proceedings of the 2017 SIAM International Conference on Data Mining.* SIAM. 2017, pp. 570–578.

[55] Wenbin Zhang. "AI fairness in practice: Paradigm, challenges, and prospects". In: *Ai Magazine* (2024).

[56] Wenbin Zhang, Tina Hernandez-Boussard, and Jeremy Weiss. "Censored fairness through awareness". In: *Proceedings of the AAAI conference on artificial intelligence.* Vol. 37. 12. 2023, pp. 14611–14619.

[57] Wenbin Zhang and Eirini Ntoutsi. "Faht: an adaptive fairness-aware decision tree classifier". In: *arXiv preprint arXiv:1907.07237* (2019).

[58] Wenbin Zhang and Jeremy C Weiss. "Fairness with censorship and group constraints". In: *Knowledge and Information Systems* 65.6 (2023), pp. 2571–2594.

[59] Wenbin Zhang and Jeremy C Weiss. "Longitudinal fairness with censorship". In: *proceedings of the AAAI conference on artificial intelligence.* Vol. 36. 11. 2022, pp. 12235–12243.

[60] Wenbin Zhang et al. "Fairness amidst non-iid graph data: A literature review". In: *arXiv preprint arXiv:2202.07170* 2 (2022).

[61] Wenbin Zhang et al. "Individual Fairness Guarantee in Learning with Censorship". In: *arXiv preprint arXiv:2302.08015* (2023).

[62] Tong Zhao et al. "Learning from counterfactual links for link prediction". In: *International Conference on Machine Learning.* PMLR. 2022, pp. 26911–26926.

[63] Yuchang Zhu et al. "FairAGG: Toward Fair Graph Neural Networks via Fair Aggregation". In: *IEEE Transactions on Computational Social Systems* (2024).

FG-SMOTE: Towards Fair Node Classification with Graph Neural Network

Zichong Wang[1], Zhipeng Yin[1], Yuying Zhang[1], Liping Yang[2], Tingting Zhang[3],
Niki Pissinou[1], Yu Cai[4], Shu Hu[5], Yun Li[6], Liang Zhao[7], and Wenbin Zhang[1*]

[1] Florida International University, Miami, FL, USA
[2] University of New Mexico, Albuquerque, NM, USA
[3] University of South Florida, Tampa, FL, USA
[4] Michigan Technology University, Houghton, MI, USA
[5] Purdue University, Indianapolis, IN, USA
[6] i4AI Ltd, London, United Kingdom
[7] Emory University, Atlanta, GA, USA

ABSTRACT

Graph generative models have become increasingly prevalent across various domains due to their superior performance in diverse applications. However, as their application rises, particularly in high-risk decision-making scenarios, concerns about their fairness are intensifying within the community. Existing graph-based generation models mainly focus on synthesizing minority nodes to enhance the node classification performance. However, by overlooking the node generation process, this strategy may intensify representational disparities among different subgroups, thereby further compromising the fairness of the model. Moreover, existing oversampling methods generate samples by selecting instances from corresponding subgroups, risking overfitting in those subgroups owing to their underrepresentation. Furthermore, they fail to account for the inherent imbalance in edge distributions among subgroups, consequently introducing structural bias when generating graph structure information. To address these challenges, this paper elucidates how existing graph-based sampling techniques can amplify real-world bias and proposes a novel framework, *Fair Graph Synthetic Minority Oversampling Technique* (FG-SMOTE), aimed at achieving a fair balance in representing different subgroups. Specifically, FG-SMOTE starts by removing the identifiability of subgroup information from node representations. Subsequently, the embeddings for simulated nodes are generated by sampling from these subgroup information desensitized node representations. Lastly, a fair link predictor is employed to generate the graph structure information. Extensive experimental evaluations on three real graph datasets show that FG-SMOTE outperforms the state-of-the-art baselines in fairness while also maintaining competitive predictive performance.

1. INTRODUCTION

Graph data is pervasive in real-world applications, such as the financial markets [48], biological networks [38], and social networks [31]. To extract node features and understand the intricate graph structures, various graph mining algorithms have been developed over the past decades [26, 37, 7, 18, 24, 54]. Among these, graph generative models have become crucial components of the graph Machine Learning (ML) framework, serving purposes such as data augmentation [5], anomaly detection [1], and recommendation [32]. For instance, in scenarios like financial fraud detection [27] and rare disease prediction [34], graph generative models can produce synthetic samples from minority community to enhance model training, thereby improving the model's ability to generalize over real-world data [56].

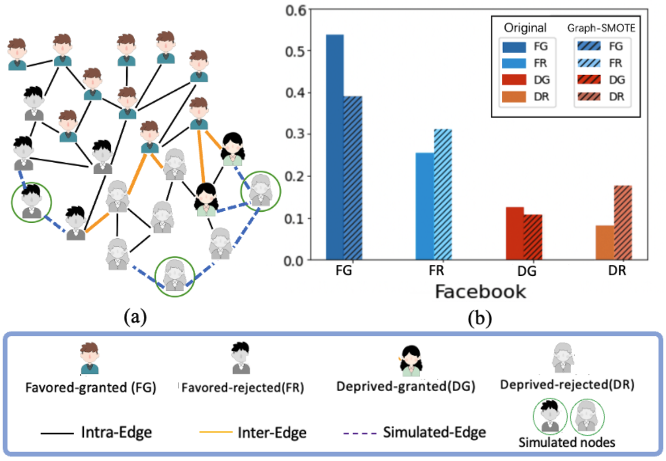


Figure 1: An illustrative example demonstrating the bias introduced by current graph generative models and its implications in the Facebook dataset.

As graph generative models become increasingly prevalent in high-risk decision-making scenarios such as credit scoring [28], recommendation [13], and healthcare [22], concerns about their fairness are intensifying [25]. Indeed, recent literature has proven graph generative models may inadvertently inherit or even exacerbate biases from real-world data [34, 34, 57]. This issue stems from distributional differences in subgroups defined by *sensitive attributes* (*e.g.*, gender or race) and labels, within graph datasets. Such dis-

*Corresponding author
Email: {ziwang, wenbin.zhang}@fiu.edu

parities can lead graph generative models to inherit or even exacerbate biases and pass them on to downstream classification tasks. In fact, despite their potential, existing graph generative models primarily focus on improving node classification performance without adequately addressing fairness implications. For instance, GraphSMOTE [56] addresses class label imbalance by generating synthetic nodes through interpolation among underrepresented groups and employs a pre-trained edge generator to establish their graph structure. Despite the remarkable effectiveness, it inadvertently intensifies disparities among subgroups, compromising the fairness of the model. To illustrate, consider credit card applications through social networking information [10] depicted in Figure 1 (a). Given the inherent higher proportion of rejected labels (*i.e.*, gray) for the deprived group (*e.g.*, females) in the dataset, GraphSMOTE generates more synthetic samples with rejected labels (surrounded by green circles) for the deprived group. Consequently, the synthetic graph depicts a reduced proportion of favorable outcomes for the deprived group, further ingraining the stereotype associating the deprived subgroup with rejected labels [45]. As a practical example, as shown in Figure 1 (b), GraphSMOTE is applied to the Facebook dataset aiming to balance the label distribution. Initially, 56.7% of samples in the deprived group had the granted label, compared to 67.8% in the favored subgroup—an 11.1% disparity. While it balances the class distribution, it further exacerbates the disparities between subgroups; the new distribution showed 36.2% favorable outcomes for the deprived subgroup and 55.6% for the favored subgroup, widening the gap to a 19.4% disparity.

Additionally, existing graph balancing techniques cannot simply be applied to simultaneously balance class labels and sensitive attribute distribution. Still, in the example shown in Figure 1 (b), the favored-granted subgroup represents 53.3% of the samples, while the deprived-granted subgroup accounts for a mere 12.4%. This evident discrepancy suggests that existing efforts to balance subgroup representation could result in the recurrent selection of samples from the deprived subgroup for synthetic sample generation. This repetitive process risks undermining sample diversity, particularly the underrepresented deprived-granted subgroup, thereby increasing the likelihood of overfitting and exacerbating existing biases.

Furthermore, existing graph generative models utilize pre-trained edge predictors to determine the graph structure for generated nodes [40]. However, biases present in the actual edge distribution can skew this process. For instance, as illustrated in Figure 1 (a), applicants are more frequently connected (*i.e.*, black line) with nodes that share the same sensitive attributes. Consequently, the graph structure generated by the model (*i.e.*, purple line) inherits and exacerbates this bias by preferentially connecting applicants with similar sensitive attributes, inadvertently introducing structural bias (*i.e.*, nodes with same sensitive attributes are excessively linked) and fostering group segregation [34].

In this paper, we investigate these challenges concerning the fairness of graph generative models. Specifically, we examine the disparities in representativeness among different subgroups and the structural biases evident in the generated graphs. We also delve into the challenges posed by sample selection in graph generative models, especially when dealing with subgroups with limited sample sizes. This domain remains largely uncharted and presents three distinct challenges: **i) Complexity of Graph Data.** Unlike tabular data, graph data does not follow the principle of Independent and Identically Distributed (I.I.D.) due to the inherent interconnections between each node and its neighbors [38]. Consequently, generating samples demands not only the creation of high-quality node features but also the accurate representation of graph structure information. **ii) Multiple sources of bias in graph data.** Biases can originate from the sensitive attribute of the nodes themselves or from the structural relationships between them [33]. Therefore, graph generative models should mitigate these multiple biases simultaneously to achieve equitable representation. **iii) Simultaneously balancing class labels and sensitive attributes.** In contrast to graph generative models driven solely by performance, fairness-aware generation necessitates simultaneous consideration of both class labels and sensitive attributes. In addition, applying conventional balancing methods directly is ineffective and may even worsen existing biases, a sophisticated design is thus essential.

To address the above challenges, we introduce a novel framework, *Fair Graph Synthetic Minority Oversampling Technique (FG-SMOTE)*, which aims at synthesizing new samples and their connections while mitigating inherent graph bias for fair node classification. *To the best of our knowledge, this is the first work that simultaneously accounts for node distribution and structural bias to generate fair graphs.* Specifically, FG-SMOTE begins by generating node embeddings for each node through aggregation. It then imposes performance and fairness constraints to ensure that the learned node embeddings remain invariant to sensitive attributes while retaining as much node information as possible. This strategy enables the following graph augmentation to select sample templates from a wider pool of sensitive information de-identified similar node embeddings. Following this, FG-SMOTE employs graph augmentation techniques, *i.e.*, cluster interpolation in the embedding space, ensuring the node generation process remains unaffected by intra-class similarity and inter-class differences. Lastly, FG-SMOTE constructs graph structural information using a fair link predictor, which is designed to prevent structural bias that might arise when generating edges based on feature similarity. Overall, integrating the strengths of both graph augmentation and graph generation, FG-SMOTE addresses the largely unattended bias issues in increasingly prevalent graph generative models, contributing to the development of a first-of-its-kind fair graph generation methodology. The key contributions of this work can be summarized as follows:

- We define a new research direction in generating fair graphs, focusing on addressing both node distributional and graph structural biases within the graphs.
- We introduce FG-SMOTE to simultaneously mitigate these inherent graph biases while generating fair graphs for equitable node classification, thereby providing essential complements to the existing literature on graph fairness.
- Extensive experimental results on three real-world graph datasets, evaluated using 11 metrics, demonstrate the effectiveness of FG-SMOTE in striking a balance between fairness and predictive performance.

2. RELATED WORK

Graph Generative Models. Existing graph generation techniques can typically be classified into two primary categories [55]: i) Node generation and ii) Edge generation. Node generation aims to extend existing oversampling methods to graph data. For instance, Oversampling [4] amplifies the representation of minority nodes by directly duplicating their embeddings from real datasets. On the other hand, GraphSMOTE [56] generates synthetic nodes for minority classes by interpolating between real embeddings, then generates the graph topology using a pre-trained edge generator. In the realm of edge generation, the goal is to capture and replicate essential structural properties of graphs. For example, Netgan [3], taking inspiration from the GAN framework [14], generates synthetic random walks while discriminating between synthetic and real random walks sampled from the input graph. GraphVAE [29] generates multiple smaller graphs and utilizes a subgraph matching algorithm to stitch them into a full-sized graph, akin to the original. GraphRNN [47] conceptualizes a graph as a sequential assembly of nodes and edges, learning this process with autoregressive models. However, their common oversight is a singular focus on performance without due consideration for fairness, which leads to the introduction of distributional or structural bias and further degrading the model's fairness.

Fairness in Graph. Existing work on graph fairness primarily focuses on improving equitable node classification [8, 41, 42, 43]. This area typically falls into two categories: individual fairness and group fairness. Specifically, individual fairness [49, 36, 12] requires that similar individuals (*e.g.*, measured by Euclidean distance) in the input space should have similar probability distributions in the output space. In contrast, group fairness [9, 15, 52] requires classifiers to have similar performance (*e.g.*, prediction accuracy) across different subgroups. More recently, there has been an emphasis on tackling structural bias in graph learning. For instance, Fairwalk [24] counters link prediction bias by adjusting the probability of random walks, ensuring equal selection chances for various subgroups. Moreover, FG^{2}AN [34] addresses biases in graph structure generation by equalizing performance across different node degrees and reducing linkage disparities among different subgroups. Nevertheless, current methodologies fall short of mitigating node distribution biases arising during the node generation.

Contrary to previous research, our approach aims to mitigate the bias in the node generation while simultaneously accounting for structural bias. Specifically, FG-SMOTE focuses on enhancing the representation of various subgroups: i) By generating samples from deprived communities, we aim to ensure consistent representation across different subgroups by a global sampling strategy; ii) By employing a fair link predictor, FG-SMOTE effectively mitigates structural biases within the graph generation model.

3. NOTIONS

In this paper, we use bold uppercase characters (*e.g.*, $\mathbf{A}$) to denote matrices, bold lowercase letters (*e.g.*, $\mathbf{s}$) to denote vectors, uppercase calligraphic characters (*e.g.*, $\mathcal{V}$) to denote sets, and normal lowercase letters (*e.g.*, s) to denote scalars. In addition, we represent the i-th row, j-th column, and (i, j)-th entry of any matrix, such as A as $A_{[i,:]}$, $A_{[:,j]}$, and $A_{i,j}$, respectively. we use lowercase bold vectors with index to represent the row vector of a matrix (*e.g.*, $a_i = A_{[i,:]}$). Moreover, we use $|\cdot|$ to denote the absolute value operator. Furthermore, we consider input graph as undirected and unweighted $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$, where $\mathcal{V}$ denote the set of nodes, $\mathcal{E}$ ($\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$) represent the set of edges, and $\mathcal{X} \in \mathbb{R}^{n \times d}$ (n = $|V|$) represent the set of node features, where x_i represents the features of the node v_i. We let $\mathbf{A}$ denote the adjacency matrix of the input graph $\mathcal{G}$, where $a_{i,j}$ takes on the value 1 if there exists an edge $i \rightarrow j$, and 0 otherwise. Meanwhile, each node v_i has a sensitive attribute, we utilize $\mathbf{S}$ $\in \{0, 1\}^{N \times 1}$ to represent the sensitive attributes, where s_i represents whether or not a given individual v_i is a member of the deprived set. Note that $s_i \in x_i$, and we let S^- denote the deprived subgroup and S^+ the favored subgroup. For every node v_i, the ego graph is G_{v_i}. The ego graph shows the direct neighbors and their connections for a specific node in the larger graph. Without restricting the generality, We let $L = \{v_1, v_2, \ldots, v_{|L|}\}$ represents the set of labeled vertices. The associated ground-truth labels are represented by Y = $\{y_1, y_2, \ldots, y_{|L|}\}$, where y_i is the label for vertex v_i. Furthermore, $U = \{v_{|L|+1}, v_{|L|+2}, \ldots, v_{|L|+|U|}\}$ represents the set of unlabeled vertices. Predicted labels for these vertices are represented as $\hat{y}$. Notably, $L \bigcup U = \mathcal{V}$. Additionally, for the convenience of the study, we assume that all sensitive attributes and labels are binary.

4. METHODOLOGY

4.1 FG-SMOTE: In a Nutshell

Figure 2 provides an overview of FG-SMOTE, showcasing its three essential modules: the Sensitive Information De-identification Module (highlighted in blue rectangle), the Data Augmentation Module (highlighted in orange rectangle), and the Fair Link Prediction Module (highlighted in green rectangle). Specifically, the Sensitive Information De-identification Module is designed to minimize the identifiability of sensitive attributes in node embeddings, while preserving as much other information as feasible by establishing three constraints (*c.f.* Section 4.2). Subsequently, the Data Augmentation Module generates samples for the underrepresented subgroups by interpolating node representations that are obtained from sensitive information de-identified embedding, ensuring balanced representation across different subgroups (*c.f.* Section 4.3). Lastly, the Fair Link Prediction Module establishes graph structure information for each synthesized sample while mitigating structural bias (*c.f.* Section 4.4). The following sections detail each of them.

4.2 Sensitive Information De-identification Module

Existing graph generative models mainly utilize node embeddings derived from encoder aggregation for oversampling, designed to balance representational differences according to class labels. However, this strategy cannot directly be applied for fair sampling, which involves distributional disparities across both sensitive attributes and labels, complicating each other. Specifically, synthetic samples are often chosen from underrepresented subgroups to balance distributional disparities. This practice can lead to the repeated generation of samples that either already exist in the training data or closely resemble existing ones, due to de-

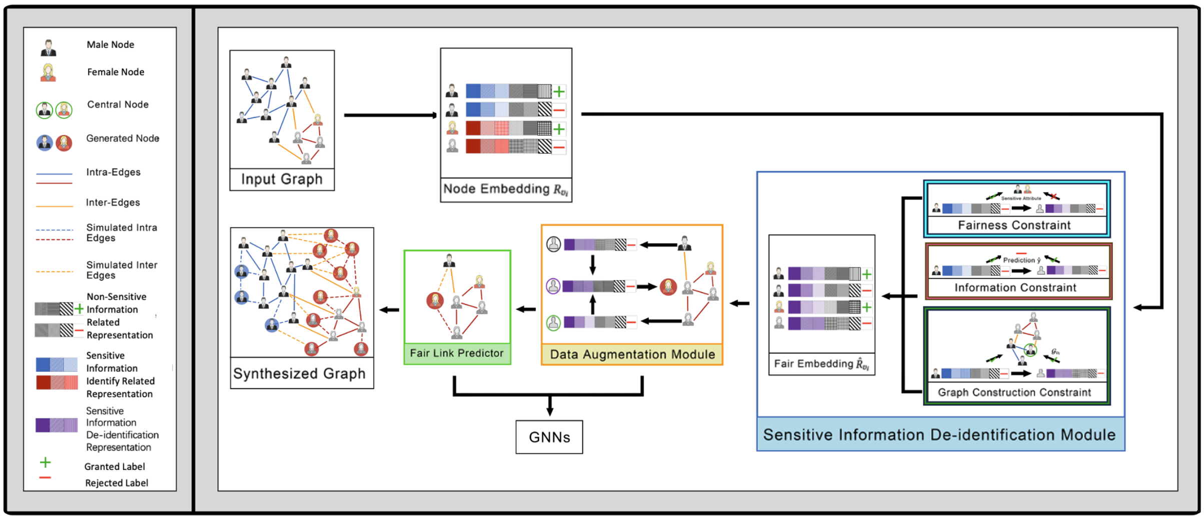


Figure 2: Overview of the proposed FG-SMOTE framework.

prived subgroups' lack of representation in the data. As a result, synthetic samples generated for these deprived subgroups lack diversity [23]. This deficiency in diversity predisposes the model to overfitting on these subgroups. To track this issue, FG-SMOTE introduces global sampling with the goal of enhancing the diversity of generated samples for underrepresented subgroups. In essence, rather than restricting the selection of sample embeddings to specific subgroup (*e.g.*, deprived rejected subgroup), we are able to select sample embeddings from the entire pool of sensitive information de-identified similar node embedding (*e.g.*, not only deprived rejected subgroup but also favored rejected subgroup), thus vastly expanding the available choices. As exemplified in the Data Augmentation Module in Figure 2, the chosen two embeddings for producing the synthetic sample (*e.g.*, 👤) is derived from distinct subgroups (*e.g.*, 👤 and 👤). This occurs because these two embeddings resemble each other after the de-identification of sensitive attribute information. To achieve this, we map each node's embedding into a new representation space. This transformation is designed to obscure any information indicating whether the individual belongs to a deprived subgroup, while preserving as much of the remaining information as possible with the following three constraints: the fairness constraint, the information constraint, and the graph construction constraint.

Fairness Constraint. The fairness constraint is designed to prevent the retention of information about the sensitive attribute, *i.e.,* favored and deprived subgroups, in node representations for fair sampling, thereby preventing the introduction of additional biases between various subgroups. Practically, within the light blue part of this module, while the original node embedding reveals information about nodes belonging to a specific group (✓), this information becomes indiscernible after applying the fairness constraint (X). Adhering to the embedding perspective, the original representation space distinctly reveals whether a node belongs to a specific subgroup. This is evident in the Node Embedding R_{v_i} in Figure 2, where sensitive information-related embeddings distinguishing between male and female are represented by red and blue colors, respectively. Subsequently, these embeddings converge towards uniform representation in the Fair Embedding $\hat{R}_{v_i}$, depicted in purple, thereby obscuring their distinct sensitive information. To achieve this, we map the node representation R_{v_i} to a new space that cannot identify whether a node belongs to a specific subgroup. Specifically, we represent each node's information in this new space using a set of prototypical probabilistic mappings; for each prototype $\rho = \hat{R}_k$, where ρ is a multinomial random variable, each value k corresponds to an instance from the intermediate set of prototypes (the dimensionality of $\hat{R}_k$ is identical to that of R_{v_i}). Hence, a node representation does not contain sensitive group information if it has the same probability of appearing in the deprived subgroup (S^-) as in the favorable subgroup (S^+), which can be mathematically represented as $P(\rho = \hat{R}_k | R \in S^+) = P(\rho = \hat{R}_k | R \in S^-)$. To efficiently map the node representation R to ρ, a natural probability mapping using the softmax function is defined as:

$$P(\rho = k|R) = \frac{\exp(-d(R, \hat{R}_k))}{\sum_{j=1}^{K} \exp(-d(R, \hat{R}_j))} \tag{1}$$

where $d(\cdot)$ is a distance metric (*e.g.*, Euclidean distance) and K denotes the number of prototypes.

Building on this, we measure the difference in the probability of each prototype in different sensitive groups. Hence, the group statistical parity difference (GSD) is formally defined as $GSD = |GP^+ - GP^-|$, where group mapping probability (GP) is defined as follows:

$$\begin{cases} GP^+ = \dfrac{1}{|v_i|^+} \displaystyle\sum_{v_i \in S^+} \sum_{k=1}^{K} P(\rho = \hat{R}_k | R_S \in S^+) \\ GP^- = \dfrac{1}{|v_i|^-} \displaystyle\sum_{v_i \in S^-} \sum_{k=1}^{K} P(\rho = \hat{R}_k | R_S \in S^-) \end{cases} \tag{2}$$

Finally, the fairness constraint is defined as follows:

$$\mathcal{L}_F = GSD + \sum_{i=1}^{n} (R_{v_i} - \hat{R}_{v_i})^2 \tag{3}$$

where $\hat{R}_{v_i}$ is the reconstructions of R_{v_i}. This constraint encourages the model to encode all information contained within the raw features except for any information that could lead to biased learning.

Information Constraint. For each node, v_i, the obtained node representation $\hat{R}_{v_i}$ should capture important node features and graph topology information to retain utility for downstream tasks. In other words, for node v_i, the model can be trained on the representations to make accurate label predictions (*i.e.,* $\hat{R}_{v_i} \to y_i$). As illustrated in the maplered part of this module, the information constraint ensures the retention of the necessary information to accurately predict (✓) the label in both the new node embedding, *i.e.,* $\hat{R}_{v_i}$, and the original node embedding, *i.e.,* R_{v_i}. Hence, the objective of the information constraint is to minimize the loss of the prediction model, as shown in Equation 4:

$$\mathcal{L}_I = \frac{1}{|\mathcal{V}_L|} \sum_{v_i \in \mathcal{V}_L} -(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)) \quad (4)$$

where y_i is the one-hot encoding of the ground-truth label of v_i. Note that current graph generative models mainly focus on R_{v_i}, thus leading to biased sampling towards deprived subgroups.

Graph Construction Constraint. For each node v_i, another objective is to ensure that its embedding accurately represents the node itself. This involves accurately reconstructing (✓) the ego-graph of node v_i, denoted as $\mathcal{G}_{v_i}$, depicted in the deep green part of this module, using the new node embedding $\hat{R}_{v_i}$. To this end, the graph construction constraint is formalized as the graph structure reconstruction loss ($\mathcal{L}_G$) as below:

$$\mathcal{L}_G = \frac{1}{|\mathcal{E}^+| + |\mathcal{E}^-|} \sum_{e_{ij} \in \mathcal{E}} L(e_{ij}, \hat{e}_{ij}) \quad (5)$$

where $\mathcal{E}^+$ and $\mathcal{E}^-$ denote the sets of sampled positive and negative edges, respectively, and $L(\cdot)$ denotes the cross-entropy loss function. In addition, e_{ij} denotes the actual connection status between nodes v_i and v_j, while the predicted probability of a connection between these nodes is given by $\hat{e}_{ij} = \sigma(\rho_i \rho_j^T)$, where $\sigma(\cdot)$ is the sigmoid function. Notable, due to the scarcity of positive edges, a negative edge is randomly selected to add to the set of negative edges for every positive edge acquired.

In essence, the introduction of the graph construction constraint serves as a precaution against noise infiltration into node representations. This ensures that the reconstructed ego-graph, $\mathcal{G}_{v_i}$, remains uncorrupted, thereby refining the quality of node generation.

4.3 Data Augmentation Module

With the fair embedding, FG-SMOTE carries out global oversampling to mitigate the rooted distributional disparities, following the general idea of SMOTE algorithm [6]. Specifically, for each node v_i, its nearest neighboring nodes bearing the same label are determined using the $Pick(\cdot)$ function, which is based on the Euclidean distance in the input space and is mathematically represented by Equation 6:

$$Pick(v_i) = \{\forall v_j \in \mathcal{G} | \text{argmin} \left\| \hat{R}_{v_i} - \hat{R}_{v_j} \right\| \} \quad \text{s.t.} \quad Y_i = Y_j \quad (6)$$

where v_i denotes the selected node, while v_j represents the neighboring node of v_i that shares the same class label but their sensitive attribute might differ.

Using the selected node embeddings of the samples and their respective nearest neighbors, we can formally define the embedding of the synthetic sample as: $R_{gen} = \alpha \hat{R}_{v_i} + (1-\alpha)\hat{R}_{v_j}$, where α is a random variable in the range $[0, 1]$, to control the similarity of synthetic node embedding is close to one instance or its close neighbor. Given that both $\hat{R}_{v_i}$ and $\hat{R}_{v_j}$ belong to the same class and are in close spatial proximity, the resulting synthetic sample R_{gen} inherits their properties and remains within the same class. The generated sample's sensitive attribute information is subsequently proportionally assigned to ensure consistent representation across different subgroups. Notably, our approach employs the SMOTE as a representative example due to its status as a foundational and widely adopted method in the realm of oversampling. However, it's important to emphasize that our methodology is versatile and can be adapted to various oversampling techniques beyond SMOTE.

4.4 Fair Link Predictor

After generating synthetic nodes to address the distributional disparities, we face another challenge: the generated nodes are not connected to the input graph $\mathcal{G}$, rendering them as isolated nodes. Consequently, it is imperative to produce graph structure information for each of these nodes to integrate them to $\mathcal{G}$. Existing methodologies harness the edge distribution of the real data to train link predictors, subsequently generating the required graph structure information. Essentially, the existence of a link between any two nodes is contingent upon their similarity according to the weighted inner product. The predicted relation ($E_{(v_i,v_j)}$) between node v_i and v_j is define as follows:

$$E_{(v_i,v_j)} = \frac{\exp(\sigma(\hat{R}_{v_i} \cdot \mathbf{Z} \cdot \hat{R}_{v_j}))}{\sum \exp(\sigma(\hat{R}_{v_i} \cdot \mathbf{Z} \cdot \hat{R}_{v_j}))} \quad (7)$$

where the parameter matrix $\mathbf{Z}$ contains the interaction between node v_i and v_j. Next, the edge generator's loss function is represented as:

$$\mathcal{L}_{IE} = \|\mathbf{E} - \mathbf{A}\|_F^2 \quad (8)$$

where $\mathbf{E}$ predicts the connections between nodes in $\mathcal{V}$.

However, this approach is performance-driven and could exacerbate the connection disparities between deprived and favored subgroups [47]. Specifically, given that most connections in the input graph $\mathcal{G}$ are intra-group connections, this leads to amplified inter-group connection disparity in the topology learned by existing graph generation models. Considering the growing practice of using social networks for credit scoring as an example: users from deprived groups may receive unjustly reduced credit scores since a significant portion of their social network consists of inter-group connections with lower credit scores. This ultimately leads to biased credit assessments against the deprived group [44].

Such a disparity can be quantified as the model performance difference between inter-group and intra-group links in $\mathcal{G}$, referred to as the consistency difference (CD), *i.e.,* CD $= |L_{inter} - L_{intra}|$, where L_{inter} represents the loss of inter-group connection, and L_{intra} denotes the loss associated with intra-group connection:

$$\begin{cases} L_{inter} = \|E_{inter} - A_{inter}\|_F^2 \\ L_{intra} = \|E_{intra} - A_{intra}\|_F^2 \end{cases} \tag{9}$$

where E_{inter} and E_{intra} denote predicted connections between inter-group and intra-group nodes, while A_{inter} and A_{intra} represent actual connections within inter-group and intra-group nodes, respectively.

Integrating these components, the loss of fair link predictor is formally defined as: $\mathcal{L}_C = \mathcal{L}_{IE} + \mathrm{CD}$

4.5 Final Joint Learning Framework

Assembling the previously discussed modules, the final objective function of FG-SMOTE, which consists of four parts and is controlled by the tunable hyperparameters a, b, and c to balance the contributions of the various elements in the overall objective function, is depicted in Equation 10:

$$\begin{aligned} \min\ \mathcal{L}_{total} &= \mathcal{L}_I + a\mathcal{L}_G + b\mathcal{L}_F + c\mathcal{L}_C \\ &= \frac{1}{|\mathcal{V}_L|} \sum_{v_i \in \mathcal{V}_L} -(y_i \log(\hat{y_i}) + (1 - y_i) \log(1 - \hat{y_i})) \\ &\quad + a \frac{1}{|\mathcal{E}^+| + |\mathcal{E}^-|} \sum_{e_{ij} \in \mathcal{E}} L(e_{ij}, \hat{e}_{ij}) + b|GP^+ - GP^-| \\ &\quad + b \sum_{i=1}^{n} (R_{v_i}, -\hat{R}_{v_i})^2 + c\|\mathbf{E} - \mathbf{A}\|_F^2 + c|L_{inter} - L_{intra}| \end{aligned} \tag{10}$$

where the first term, $\mathcal{L}_I$, aims to minimize the prediction loss; the second term, $\mathcal{L}_G$, design to minimize the reconstruction loss for the node representations R with the function $L(\cdot)$ denoting cross-entropy; The third term, $\mathcal{L}_F$, aiming to desensitize the representation R, thereby enable global sampling to mitigate node representation bias; The last term, $\mathcal{L}_C$, aims to promote the fairness of the edge predictor and avoid introducing structural bias.

5. EXPERIMENT

5.1 Experimental Settings

Datasets. Three real-world graph datasets with varying characteristics are used for thorough evaluations: i) The **Facebook** dataset [20] contains the Ego Network of social networks, where each node represents a user, and if there is an edge between two nodes, it means that the two users are Facebook friends with each other. The sensitive attribute is gender, and the aim is to predict whether users are in the same social circle. ii) The **Pokec-z** [30] is derived from a popular social network in Slovakia. Nodes denote users with features such as gender, age, interest, etc. Edge represents the friendship between users. Considering the region as the sensitive attribute, the task is to predict the working field of the users. iii) The **Credit** dataset [20] contains payment default information for each individual. In this dataset, each node represents an individual, and each edge between the two nodes indicates a similarity in their payment methods. The sensitive attribute is age, with the aim of predicting whether their default payment is to use a credit card. The detailed statistical information for each graph dataset is shown in Table 1.

Table 1: Summary of the datasets used in the experiments.

Dataset	Facebook	Pokec-z	Credit	Bail
Vertices	1,034	67,796	30,000	18,876
Edges	26,749	651,856	137,377	311,870
Average Degree	51.7	19.2	10	34
Sensitive Attribute	Gender	Region	Age	Race

Evaluation Metrics. Eleven metrics encompassing node classification performance, graph generation quality, and their respective fairness considerations are employed for comprehensive evaluation:

i) Node Classification Metrics. Three metrics are adopted to evaluate the node classification performance, *i.e.,* accuracy, F1-Score, and AUC. For all of them, higher values correspond to better performance. In terms of fairness, the evaluation is based on Statistical Parity Differences (SPD) [19] and Equal Opportunity Differences (EOD) [15]. For both of them, an absolute value close to 0 indicates optimal fairness, while larger values denote more significant discrimination.

ii) Edge Generation Evaluation Metrics. Three metrics are employed to evaluate the quality of the generated graph. Specifically, the difference between the following properties of the generated graph and the original graph are measured [58]: 1) Mean Degree Difference (MD): The average node degree; 2) Edge Distribution Entropy Difference (EDED): The relative edge distribution entropy of $\mathcal{G}$; 3) Gini Difference (GD): The Gini coefficient of the degree distribution. Moreover, drawing from prior work [34], three fair edge generation metrics are adopted to gauge disparities when generating edges for favored and deprived subgroups: 1) Average Degree Difference (ADD): Evaluates the disparity in network clustering difference between deprived and favored node subgroups; 2) Equal Edge Distribution Entropy (EEDE): Quantifies the disparity between the relative edge distribution entropy of the favored and deprived subgroups; 3) Equal Gini (EG): Evaluates the disparity in the Gini coefficient of the degree distribution across different subgroups. For all of these six metrics, smaller values indicate better predictive performance and fairness.

Baselines. To evaluate the efficacy of FG-SMOTE, nine state-of-the-art graph generative models from different perspectives are compared: Oversampling [4], Embed-SMOTE [2], GraphSMOTE [56], FairGNN [11], Graphair [21], FairAGG [59], RFCGNN [39], FDGNN [35], and FG^{2}AN [34]. Specifically, the first three models are performance-driven, focusing on oversampling the minority class to enhance the classifier's performance. On the other hand, the subsequent six are fairness-aware by design. Among them, **FairGNN** employs adversarial learning to cultivate GNNs adhering to group fairness criteria; **Graphhair** aims to generate fair graph data using adver-

Table 2: Comparison results of FG-SMOTE with baseline methods across real-world datasets. In each row, the best result is indicated in bold, while the runner-up result is marked with an underline.

Dataset	Methods / Metrics	Oversampling	Embed-SMOTE	GraphSMOTE	FairGNN	Graphair	FairAGG	RFCGNN	FDGNN	**FG-SMOTE**
Facebook	**Accuracy** (↑)	0.721 ± 0.013	**0.747 ± 0.017**	0.743 ± 0.015	0.667 ± 0.024	0.563 ± 0.015	0.668 ± 0.021	0.663 ± 0.019	0.713 ± 0.017	0.744 ± 0.023
	F1-Score (↑)	0.729 ± 0.024	0.730 ± 0.019	0.743 ± 0.024	0.673 ± 0.021	0.619 ± 0.022	0.683 ± 0.022	0.692 ± 0.028	0.724 ± 0.031	**0.749 ± 0.015**
	AUC(↑)	0.769 ± 0.055	0.783 ± 0.045	**0.788 ± 0.067**	0.687 ± 0.031	0.581 ± 0.003	0.658 ± 0.031	0.745 ± 0.033	0.756 ± 0.041	0.782 ± 0.004
	SPD(↓)	0.157 ± 0.065	0.131 ± 0.042	0.0116 ± 0.045	0.083 ± 0.027	0.043 ± 0.021	0.045 ± 0.024	0.041 ± 0.023	0.039 ± 0.015	**0.037 ± 0.019**
	EOD(↓)	0.132 ± 0.032	0.114 ± 0.027	0.103 ± 0.048	0.067 ± 0.024	0.027 ± 0.017	0.031 ± 0.011	0.025 ± 0.033	0.023 ± 0.019	**0.022 ± 0.012**
Pokec-z	**Accuracy** (↑)	0.728 ± 0.015	**0.755 ± 0.008**	0.740 ± 0.012	0.724 ± 0.021	0.655 ± 0.004	0.735 ± 0.032	0.698 ± 0.091	0.724 ± 0.013	0.739 ± 0.013
	F1-Score (↑)	0.791 ± 0.012	0.817 ± 0.012	**0.828 ± 0.015**	0.687 ± 0.033	0.647 ± 0.019	0.716 ± 0.005	0.776 ± 0.029	0.789 ± 0.032	0.824 ± 0.022
	AUC(↑)	0.723 ± 0.015	0.756 ± 0.025	0.788 ± 0.017	0.761 ± 0.011	0.653 ± 0.013	0.731 ± 0.013	0.738 ± 0.033	0.747 ± 0.005	**0.789 ± 0.004**
	SPD(↓)	0.116 ± 0.032	0.103 ± 0.028	0.098 ± 0.026	0.038 ± 0.022	0.035 ± 0.008	**0.022 ± 0.014**	0.031 ± 0.007	0.030 ± 0.017	0.024 ± 0.007
	EOD(↓)	0.101 ± 0.031	0.095 ± 0.026	0.081 ± 0.017	0.033 ± 0.029	0.032 ± 0.006	0.028 ± 0.011	0.027 ± 0.013	**0.022 ± 0.011**	0.025 ± 0.013
Credit	**Accuracy** (↑)	0.755 ± 0.017	0.774 ± 0.013	0.781 ± 0.017	0.687 ± 0.012	0.531 ± 0.024	0.653 ± 0.034	0.735 ± 0.017	0.736 ± 0.020	**0.801 ± 0.022**
	F1-Score (↑)	0.802 ± 0.028	0.834 ± 0.011	**0.871 ± 0.018**	0.783 ± 0.043	0.728 ± 0.072	0.747 ± 0.042	0.849 ± 0.049	0.861 ± 0.048	0.868 ± 0.052
	AUC(↑)	0.734 ± 0.015	0.741 ± 0.015	**0.771 ± 0.027**	0.711 ± 0.074	0.758 ± 0.047	0.721 ± 0.022	0.743 ± 0.033	0.747 ± 0.031	0.763 ± 0.014
	SPD(↓)	0.161 ± 0.035	0.117 ± 0.037	0.153 ± 0.037	0.123 ± 0.036	0.085 ± 0.034	0.074 ± 0.047	0.074 ± 0.047	0.056 ± 0.024	**0.051 ± 0.014**
	EOD(↓)	0.127 ± 0.035	0.103 ± 0.047	0.108 ± 0.047	0.115 ± 0.042	0.088 ± 0.035	0.056 ± 0.021	0.064 ± 0.016	0.047 ± 0.016	**0.044 ± 0.013**

sarial learning to deceive a discriminator to achieve fairness; **FairAGG** implements a fair aggregation scheme based on the Shapley value to ensure group fairness; **RFCGNN** learns a fair node representation by identifying counterfactual instances and sensitive attribute-related information masking; **FDGNN** utilizes counterfactual samples to learn disentangled node representation to mitigate the multi-source biases, and **FG^{2}AN** is designed to foster fair edge generation for equitable graph structure information.

5.2 Implementation Details

For FG-SMOTE, we employ the Adam optimizer [16] with a learning rate of $lr = 0.001$, setting $epochs = 1000$ and a weight decay of 1×10^{-5}. For all baseline methods except FG^{2}AN, we use a 1-layer GCN [17] with 16 hidden dimensions as the model backbone, coupled with a linear layer for classification. We ensure fairness and optimal performance across all models by tuning the hyperparameters based on each method's performance on the validation set. For the FG^{2}AN method, we followed the author's instructions and configured the number of transformer heads to 4, set the learning rate to $lr = 0.01$, and the walk length to $T = 10$. All models are implemented with PyTorch and PyTorch-Geometric.

5.3 Experiment Results

The following five research questions are investigated to comprehensively evaluate FG-SMOTE.

RQ1: How does FG-SMOTE perform? FG-SMOTE is benchmarked against eight state-of-the-art baselines using four real-world graph datasets. Each experiment is repeated 10 times, with the average performance reported. To ensure a fair comparison, the selection of hyperparameters for all methods was optimized based on their performance on a validation set, with the results summarized in Table 2. Note that FG^{2}AN generates graph structure only and is thus excluded from RQ1. As we can see, FG-SMOTE demonstrates remarkable advantages in both fairness and performance. Out of 9 performance comparisons, FG-SMOTE achieved the highest ranking in 3 and was the second-highest in 4. In the 6 fairness comparisons, FG-SMOTE was ranked first 4 times and second twice. Specifically, FG-SMOTE considers fairness and representativeness simultaneously to balance the distribution of class labels and sensitive attributes while adopting a global sampling strategy that effectively sidesteps overfitting risks. Hence, FG-SMOTE emerges as an equitable option for socially sensitive graph-based applications.

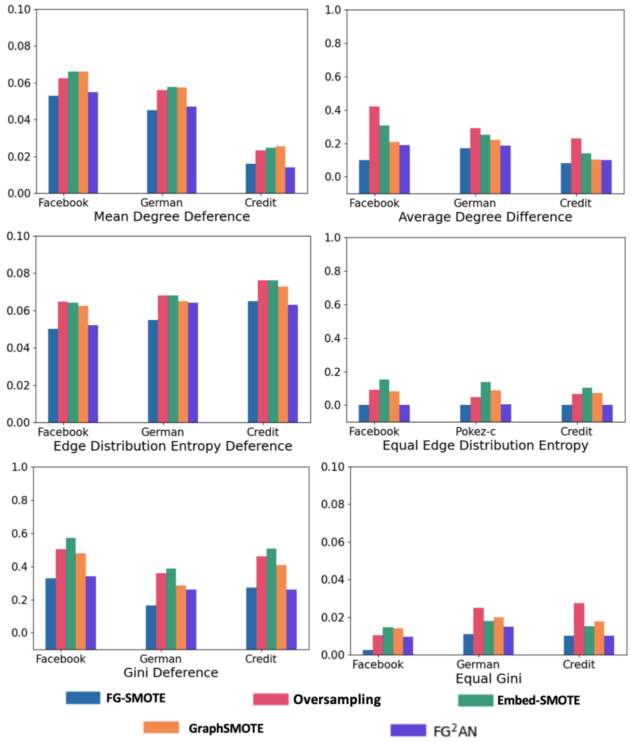


Figure 3: Comparative results of FG-SMOTE and baselines generated graph quality.

RQ2: What is the quality of graphs generated by FG-SMOTE? FG-SMOTE is compared with three graph sampling techniques and a fair graph structure generation method with the results shown in Figure 3. Note that FairGNN, Graphair, FairAGG, RFCGNN, and FDGNN are designed for classification tasks and do not generate nodes or edges, rendering a comparison with them infeasible. Overall, FG-SMOTE generates high-quality graphs while ensuring that the generated graph structure information is devoid of discrimination information. This is because existing graph sampling techniques overlook the disparity in edge distribution when generating graph structure information, resulting in stronger intra-connections and group isolation. In contrast, FG-SMOTE, equipped with a fair link predictor, effectively addresses both intra-group and inter-group distribution disparities while ensuring performance.

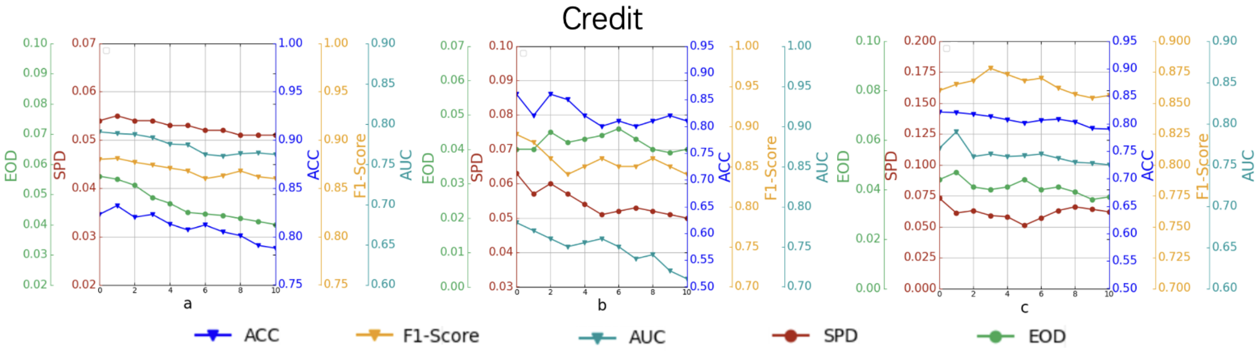


Figure 4: Exploring hyperparameters study results in the Credit dataset.

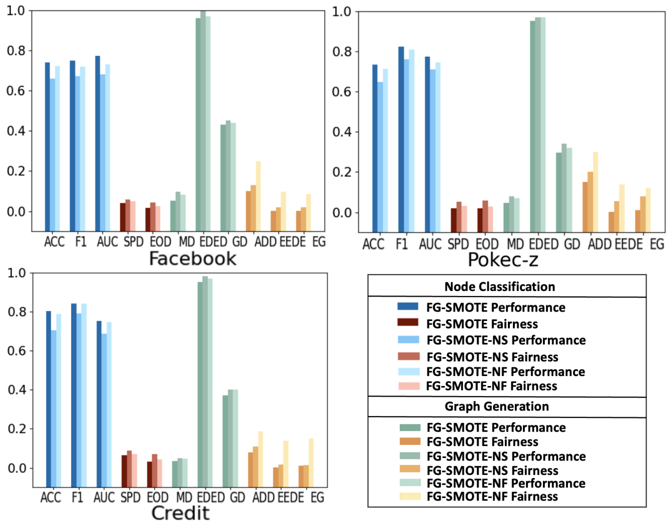


Figure 5: Ablation study results for FG-SMOTE, FG-SMOTE-NS, and FG-SMOTE-NF.

RQ3: What is the impact of each module in the FG-SMOTE framework on its overall performance and fairness? Various ablation studies are conducted to answer RQ3. First, the FG-SMOTE-NS variant is created by removing the Sensitive Information De-identification Module to evaluate the effectiveness of this module. In this setting, node embeddings obtained from a standard GNN encoder are directly used for data augmentation aimed at enhancing the representation of under-represented subgroups. The results are shown in Figure 5, highlighting a deterioration in both performance and fairness metrics compared to the FG-SMOTE. This drop is due to the limitation on synthetic sample selection, which is limited to corresponding subgroups, leading to reduced sample diversity. This limitation increases the model's susceptibility to overfitting, negatively impacting generalizability, performance, fairness, and ultimately, the quality of the constructed graph.

Next, the Fair Link Prediction Module's impact was assessed by evaluating the FG-SMOTE-NF variant, which excludes this module. As we can see from the results shown in Figure 5, FG-SMOTE-NF shows a decrease in fairness, demonstrating the importance of the Fair Link Prediction Module. Without this module, the model generates a biased graph structure, causing groups sharing the same sensitive attributes to be linked more closely. As a result, the model's node predictions are more influenced by sensitive attributes, leading to discriminatory decisions. Additionally, the lack of differentiation between inter- and intra-group connection losses in the edge prediction exacerbates intra-group connectivity, intensifying structural bias in the generated graph.

RQ4: How do hyperparameters affect the performance and fairness of FG-SMOTE? Three hyperparameters a, b, and c within FG-SMOTE are investigated for their roles in governing node reconstruction performance, fair node generation, and the structural bias of the framework, respectively. Figure 4 delineates the effects of modulating these hyperparameters, varying as $\{0, 1, \ldots, 10\}$, on both the performance and fairness of FG-SMOTE. Specifically, an increase in a will increase model fairness but at the cost of some predictive performance degradation. This is attributed to the enhanced ability of node representations to learn the graph's structural information, which, in turn, reduces bias arising from unequal neighbor distribution. In addition, an increase of b results in a minor decrease in performance while fairness improves. Specifically, a higher b value amplifies the framework's capability to desensitize node embeddings, thus curtailing bias introduced during node generation. As for c, increasing its value initially improves fairness, and beyond a certain threshold, fairness declines while the impact on performance remains minimal. This is due to enhancing c bolsters the model's proficiency in accurately generating graph structural, thus circumventing the infusion of structural bias.

6. CONCLUSION

This paper is driven by growing concerns over discriminatory practices in graph generation models. Unlike existing techniques, the proposed FG-SMOTE focuses on addressing node distribution and structural biases prevalent in real-world graph applications. It employs constraints related to fairness, performance, and graph structure reconstruction to ensure both sensitive information de-identification and preservation of critical prediction information. Moreover, by employing global sampling, FG-SMOTE enhances the diversity of generated nodes, ensuring a balanced representation of class labels and sensitive attributes. Experimental results on four real graphs validate FG-SMOTE's effectiveness in prediction performance and fairness. In addition, this paper explores a new research direction and lays the groundwork for future advancements in fair graph generation.

Acknowledgement

This work was supported in part by the National Science Foundation (NSF) under Grant No. 2245895.

7. REFERENCES

[1] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. "RTM: Laws and a recursive generator for weighted time-evolving graphs". In: *2008 Eighth IEEE International Conference on Data Mining*. IEEE. 2008, pp. 701–706.

[2] Shin Ando and Chun Yuan Huang. "Deep oversampling framework for classifying imbalanced data". In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 10*. Springer. 2017, pp. 770–785.

[3] Aleksandar Bojchevski et al. "Netgan: Generating graphs via random walks". In: *International conference on machine learning*. PMLR. 2018, pp. 610–619.

[4] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. "A systematic study of the class imbalance problem in convolutional neural networks". In: *Neural networks* 106 (2018), pp. 249–259.

[5] Deepayan Chakrabarti and Christos Faloutsos. "Graph mining: Laws, generators, and algorithms". In: *ACM computing surveys (CSUR)* 38.1 (2006), 2–es.

[6] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[7] Flavio Chierichetti et al. "Fair clustering through fairlets". In: *Advances in neural information processing systems* 30 (2017).

[8] Manvi Choudhary, Charlotte Laclau, and Christine Largeron. "A survey on fairness for machine learning on graphs". In: *arXiv preprint arXiv:2205.05396* (2022).

[9] Alexandra Chouldechova. "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments". In: *Big data* 5.2 (2017), pp. 153–163.

[10] Zhibo Chu, Zichong Wang, and Wenbin Zhang. "Fairness in Large Language Models: A Taxonomic Survey". In: *ACM SIGKDD Explorations Newsletter, 2024* (2024), pp. 34–48.

[11] Enyan Dai and Suhang Wang. "Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information". In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 2021, pp. 680–688.

[12] Cynthia Dwork et al. "Fairness through awareness". In: *Proceedings of the 3rd innovations in theoretical computer science conference*. 2012, pp. 214–226.

[13] Wenqi Fan et al. "Graph neural networks for social recommendation". In: *The world wide web conference*. 2019, pp. 417–426.

[14] Ian Goodfellow et al. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[15] Moritz Hardt, Eric Price, and Nati Srebro. "Equality of opportunity in supervised learning". In: *Advances in neural information processing systems* 29 (2016).

[16] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[17] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[18] Matthäus Kleindessner et al. "Guarantees for spectral clustering with fairness constraints". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3458–3467.

[19] Tai Le Quy et al. "A survey on datasets for fairness-aware machine learning". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 12.3 (2022), e1452.

[20] Jure Leskovec and Julian Mcauley. "Learning to discover social circles in ego networks". In: *Advances in neural information processing systems* 25 (2012).

[21] Hongyi Ling et al. "Learning fair graph representations via automated data augmentations". In: *International Conference on Learning Representations (ICLR)*. 2023.

[22] Ziad Obermeyer et al. "Dissecting racial bias in an algorithm used to manage the health of populations". In: *Science* 366.6464 (2019), pp. 447–453.

[23] Joonhyung Park, Jaeyun Song, and Eunho Yang. "Graphens: Neighbor-aware ego network synthesis for class-imbalanced node classification". In: *International Conference on Learning Representations*. 2021.

[24] Tahleen Rahman et al. "Fairwalk: Towards fair graph embedding". In: (2019).

[25] Nripsuta Ani Saxena, Wenbin Zhang, and Cyrus Shahabi. "Missed opportunities in fair AI". In: *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM. 2023, pp. 961–964.

[26] Prithviraj Sen et al. "Collective classification in network data". In: *AI magazine* 29.3 (2008), pp. 93–93.

[27] Shubhranshu Shekhar, Neil Shah, and Leman Akoglu. "Fairod: Fairness-aware outlier detection". In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 2021, pp. 210–220.

[28] Valentina Shumovskaia et al. "Linking bank clients using graph neural networks powered by rich transactional data: Extended abstract". In: *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. 2020, pp. 787–788.

[29] Martin Simonovsky and Nikos Komodakis. "Graphvae: Towards generation of small graphs using variational autoencoders". In: *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*. Springer. 2018, pp. 412–422.

[30] Lubos Takac and Michal Zabovsky. "Data analysis in public social networks". In: *International scientific conference and international workshop present day trends of innovations*. Vol. 1. 6. 2012.

[31] Huaiyu Wan et al. "Aminer: Search and mining of academic social networks". In: *Data Intelligence* 1.1 (2019), pp. 58–76.

[32] Xiang Wang et al. "Neural graph collaborative filtering". In: *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval.* 2019, pp. 165–174.

[33] Zichong Wang and Wenbin Zhang. "Group Fairness with Individual and Censorship Constraints". In: *27th European Conference on Artificial Intelligence.* 2024.

[34] Zichong Wang et al. ": Fairness-aware graph generative adversarial networks". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer. 2023, pp. 259–275.

[35] Zichong Wang et al. "Advancing Graph Counterfactual Fairness through Fair Representation Learning". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer Nature Switzerland. 2024, pp. 40–58.

[36] Zichong Wang et al. "Graph Fairness via Authentic Counterfactuals: Tackling Structural and Causal Challenges". In: *ACM SIGKDD Explorations Newsletter, 2025* (2025).

[37] Zichong Wang et al. "History, Development, and Principles of Large Language Models-An Introductory Survey". In: *AI and Ethics, 2024* (2024).

[38] Zichong Wang et al. "Individual Fairness with Group Awareness Under Uncertainty". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer Nature Switzerland. 2024, pp. 89–106.

[39] Zichong Wang et al. "Mitigating multisource biases in graph neural networks via real counterfactual samples". In: *2023 IEEE International Conference on Data Mining (ICDM).* IEEE. 2023, pp. 638–647.

[40] Zichong Wang et al. "Preventing Discriminatory Decision-making in Evolving Data Streams". In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT).* 2023.

[41] Zichong Wang et al. "Toward Fair Graph Neural Networks via Real Counterfactual Samples". In: *Knowledge and Information Systems* (2024), pp. 1–25.

[42] Zichong Wang et al. "Towards Fair Graph Pooling with Group and Individual Awareness". In: *proceedings of the AAAI conference on artificial intelligence.* 2025.

[43] Zichong Wang et al. "Towards fair machine learning software: Understanding and addressing model bias through counterfactual thinking". In: *arXiv preprint arXiv:2302.08018* (2023).

[44] Yanhao Wei et al. "Credit scoring with social network data". In: *Marketing Science* 35.2 (2016), pp. 234–258.

[45] Depeng Xu et al. "Fairgan: Fairness-aware generative adversarial networks". In: *2018 IEEE International Conference on Big Data (Big Data).* IEEE. 2018, pp. 570–575.

[46] Zhipeng Yin, Zichong Wang, and Wenbin Zhang. "Improving Fairness in Machine Learning Software via Counterfactual Fairness Thinking". In: *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings.* 2024, pp. 420–421.

[47] Jiaxuan You et al. "Graphrnn: Generating realistic graphs with deep auto-regressive models". In: *International conference on machine learning.* PMLR. 2018, pp. 5708–5717.

[48] Si Zhang et al. "Hidden: hierarchical dense subgraph detection with application to financial fraud detection". In: *Proceedings of the 2017 SIAM International Conference on Data Mining.* SIAM. 2017, pp. 570–578.

[49] Wenbin Zhang. "AI fairness in practice: Paradigm, challenges, and prospects". In: *Ai Magazine* (2024).

[50] Wenbin Zhang, Tina Hernandez-Boussard, and Jeremy Weiss. "Censored fairness through awareness". In: *Proceedings of the AAAI conference on artificial intelligence.* Vol. 37. 12. 2023, pp. 14611–14619.

[51] Wenbin Zhang and Eirini Ntoutsi. "Faht: an adaptive fairness-aware decision tree classifier". In: *arXiv preprint arXiv:1907.07237* (2019).

[52] Wenbin Zhang and Jeremy C Weiss. "Fairness with censorship and group constraints". In: *Knowledge and Information Systems* 65.6 (2023), pp. 2571–2594.

[53] Wenbin Zhang and Jeremy C Weiss. "Longitudinal fairness with censorship". In: *proceedings of the AAAI conference on artificial intelligence.* Vol. 36. 11. 2022, pp. 12235–12243.

[54] Wenbin Zhang et al. "Fairness amidst non-iid graph data: A literature review". In: *arXiv preprint arXiv:2202.07170* 2 (2022).

[55] Wenbin Zhang et al. "Individual Fairness under Uncertainty". In: *26th European Conference on Artificial Intelligence.* 2023, pp. 3042–3049.

[56] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. "Graphsmote: Imbalanced node classification on graphs with graph neural networks". In: *Proceedings of the 14th ACM international conference on web search and data mining.* 2021, pp. 833–841.

[57] Lecheng Zheng et al. "Fairgen: Towards fair graph generation". In: *2024 IEEE 40th International Conference on Data Engineering (ICDE).* IEEE. 2024, pp. 2285–2297.

[58] Dawei Zhou et al. "A data-driven graph generative model for temporal interaction networks". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2020, pp. 401–411.

[59] Yuchang Zhu et al. "FairAGG: Toward Fair Graph Neural Networks via Fair Aggregation". In: *IEEE Transactions on Computational Social Systems* (2024).

Time Series Forecasting with LLMs: Understanding and Enhancing Model Capabilities

Hua Tang[2,*], Chong Zhang[3,*], Mingyu Jin[1], Qinkai Yu[3], Zhenting Wang[1],
Xiaobo Jin[5], Yongfeng Zhang[1], Mengnan Du[4].

[1]Rutgers University, [2]Shanghai Jiaotong University, [3]University of Liverpool,
[4]New Jersey Institute of Technology, [5]Xi'an Jiaotong-Liverpool University.

ABSTRACT

Large language models (LLMs) have been applied in many fields and have developed rapidly in recent years. As a classic machine learning task, time series forecasting has recently been boosted by LLMs. Recent works treat large language models as *zero-shot* time series reasoners without further fine-tuning, which achieves remarkable performance. However, some unexplored research problems exist when applying LLMs for time series forecasting under the zero-shot setting. For instance, the LLMs' preferences for the input time series are less understood. In this paper, by comparing LLMs with traditional time series forecasting models, we observe many interesting properties of LLMs in the context of time series forecasting. First, our study shows that LLMs perform well in predicting time series with clear patterns and trends but face challenges with datasets lacking periodicity. This observation can be explained by the ability of LLMs to recognize the underlying period within datasets, which is supported by our experiments. In addition, the input strategy is investigated, and it is found that incorporating external knowledge and adopting natural language paraphrases substantially improve the predictive performance of LLMs for time series. Our study contributes insight into LLMs' advantages and limitations in time series forecasting under different conditions.

1. INTRODUCTION

Recently, large language models (LLMs) have been widely used and have achieved promising performance across various domains, such as health management, customer analysis, and text feature mining [1–4]. Time series forecasting requires extrapolation from sequential observations. Language models are designed to discern intricate concepts within temporally correlated sequences and intuitively appear well-suited for this task. Hence, some preliminary studies apply LLMs to time series forecasting tasks [5–7].

However, the application of LLMs for time series forecasting is still in its early stage, and the boundaries of this research area are not yet well defined. There are many unexplored problems in this field. For example, existing research lacks exploration into how the performance of LLMs varies when faced with different types of time series inputs. This includes the effectiveness gap for LLMs in predicting data with seasonal and trending patterns versus data without such patterns.

To fill this research gap, in this paper, we focus on LLMs' preferences for the input time series in time series forecasting under the zero-shot prompting setting. Experiments on both real and synthesized datasets show that LLMs perform better in time series with higher trend or seasonal strengths. Our observations also reveal that LLMs perform worse when there are multiple periods within datasets, which may be attributed to the fact that LLMs cannot capture distinct periods within those datasets. To further discern the LLMs' preferences for the specific input data segments, we design counterfactual experiments involving systematic permutations of input sequences. The findings suggest that LLMs are particularly sensitive to the segment of input sequences closest to the target output.

Based on the above findings, we want to explore why LLMs forecast well on datasets with higher seasonal strengths. To this end, we require LLMs to tell the period of the datasets through multiple runs. We find that LLMs can mostly recognize the underlying period of a dataset. This can explain the findings of why large language models can forecast time series with high trends or seasonal intensities well since they can obtain the seasonal pattern inside the datasets.

In light of the above-mentioned findings, we are interested in how to leverage these insights to further improve model performance. To address this, we propose two simple techniques to enhance model performance: incorporating external human knowledge and converting numerical sequences into natural language counterparts. Incorporating supplementary information enables large language models to more effectively grasp the periodic nature of time series data, moving beyond a mere emphasis on the tail of the time series. Transforming numerical data into a natural language format enhances the model's ability to comprehend and reason, also serving as a beneficial approach. Both approaches improve model performance and contribute to our understanding of LLMs in time series forecasting. The workflow is illustrated in Figure 1.

The key contributions are as follows:

- We investigate the preferences for the input sequences in LLMs in time series forecasting tasks. Our analysis has revealed that LLMs significantly outperform traditional time series forecasting methods without the need for additional fine-tuning. Interestingly, LLMs display superior predictive capabilities when dealing with datasets that have higher trends and seasonal strengths.
- We require LLMs to identify the periodicity of datasets across multiple iterations. Our observations indicate that

LLMs can effectively recognize the inherent periodic patterns within datasets. This observation answers the question of why LLMs perform well in forecasting time series with higher seasonal strengths, as they can capture the seasonal patterns inherent in the data.

- We propose two simple techniques to improve model performance and find that both incorporating external human knowledge into input prompts and paraphrasing input sequences to natural language substantially improve the performance of LLMs in time series forecasting.

2. PRELIMINARIES

2.1 Large Language Model

We use LLMs as a zero-shot learner for time series forecasting by treating numerical values as text sequences. In this paper, we investigate three close source LLMs, including GPT-3.5-turbo, GPT-4-turbo, and Gemini-1.0-Pro, and one open-source LLMs, i.e., llama-2-13B. The success of LLMs in time series forecasting can significantly depend on correct pre-processing and handling of the data [5]. We followed the pre-processing approach of Gruver [5] and this process involves the following few steps.

Input Pre-processing. In this phase of time series forecasting with LLMs, we perform two pre-processing steps. First, numerical values are transformed into strings, a crucial step that significantly influences the model's comprehension and data processing. For instance, a series like 0.123, 1.23, 12.3, 123.0 is reformatted to "1 2, 1 2 3, 1 2 3 0, 1 2 3 0 0", introducing spaces between digits and commas to delineate time steps, while decimal points are omitted to save token space. Second, tokenization is equally important, shaping the model's pattern recognition capabilities. Unlike traditional methods such as byte-pair encoding (BPE) [8], which can disrupt numerical coherence, we use spacing digits which ensures individual tokenization, enhancing pattern discernment. Third, rescaling is employed to efficiently utilize tokens and manage large inputs by adjusting values so that a specific percentile aligns to 1. This facilitates the model's exposure to varying digit counts and supports the generation of larger values, a testament to the nuanced yet critical nature of data preparation in leveraging LLMs for time series analysis.

2.2 Time Series Forecasting

In the context of time-series forecasting, the primary goal is to predict the values for the next H steps based on observed values from the preceding K steps, which is mathematically expressed as:

$$\hat{X}_t, ..., \hat{X}_{t+H-1} = F(X_{t-1}, ..., X_{t-K}; V; \lambda) \tag{1}$$

Here, $\hat{X}_t, ..., \hat{X}_{t+H-1}$ represent the H-step estimation given the previous K-step values $X_{t-1}, ..., X_{t-K}$. λ denotes the trained parameters from the model F, and V denotes the prompt or any other information used for inference. In this paper, we focus predominantly on univariate time series forecasting to investigate the preference and performance of LLMs in univariate time series forecasting under the zero-shot setting.

Motivated by interpretability requirements in real-world scenarios, time series can often be decomposed into the trend component, the seasonal component, and the residual component through the addictive model [9]. The trend component captures the hidden long-term changes in the data, such as the linear or exponential pattern. The seasonal component captures the repeating variation in the data, and the residual component captures the remaining variation in the data after removing the trend and seasonal components. This decomposition offers a method to quantify the properties of time series, which is detailed in subsection 3.2.

Datasets. In this study, we primarily use Darts [10], a benchmark univariate dataset widely recognized in deep learning research, along with many baseline methods. Darts consists of eight real univariate time series datasets, including those with clear patterns, such as the AirPassengerDataset, and irregular datasets, such as the SunspotsDataset. Besides, we employ some other commonly used datasets, such as US Births Dataset [11], TSMC-Stock and Turkeypower datasets [5] and ETT [12] in Sections 5.2 and 5.3 to demonstrate the effectiveness of our proposed methods. A full description of those datasets can be seen in Section 5.1.

Evaluation Metrics. In this paper, we evaluate model performance with three metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). These metrics are defined as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{4}$$

where y_i denotes the true value, $\hat{y}_i$ represents the predicted value, and n is the sample size.

3. WHAT ARE LLMS' PREFERENCES IN TIME SERIES FORECASTING?

To explore the preference of LLMs, we first quantify the properties of the input time series to investigate the LLMs' preferences for time series. Then, to further emphasize our findings, we evaluate the importance of different segments of the input sequence by adding Gaussian noise to the original time series.

3.1 Analyzing Method

We first compare the performance between LLMs and traditional time series forecasting methods, as shown in Table 2. It is shown that LLMs perform better within most datasets. GPT-4-turbo and Llama-2 perform relatively well on the AirPassengerdataset and the AusBeerdataset with low MAPE. Gemini outperforms GPT-3.5-turbo on time series forecasting and outperforms GPT-4-turbo on some datasets but is on par with GPT-4-turbo overall.

To understand the preferences of the LLMs, we compare our framework using various foundational models, such as GPT-4-turbo and GPT-3.5-turbo, with traditional methods. We also design experiments on synthesized datasets to validate our findings and analyze the impact of the multiple periods. To quantify the LLMs' preferences towards time series, following

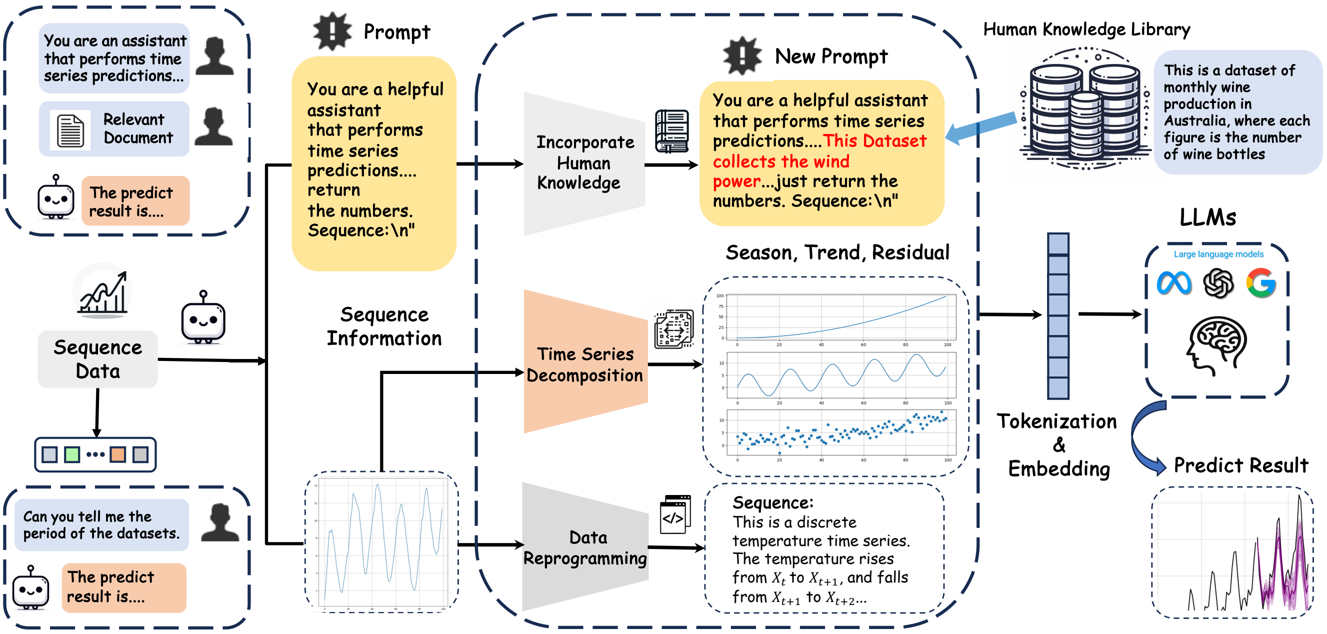

You are an assistant that performs time series predictions...
Relevant Document
The predict result is....
Sequence Data
Can you tell me the period of the datasets.
The predict result is....
Prompt
You are a helpful assistant that performs time series predictions.... return the numbers. Sequence:\n"
Sequence Information
Incorporate Human Knowledge
Time Series Decomposition
Data Reprogramming
New Prompt
You are a helpful assistant that performs time series predictions....This Dataset collects the wind power...just return the numbers. Sequence:\n"
Human Knowledge Library
This is a dataset of monthly wine production in Australia, where each figure is the number of wine bottles
Season, Trend, Residual
Sequence: This is a discrete temperature time series. The temperature rises from X_t to X_{t+1}, and falls from X_{t+1} to X_{t+2}...
Tokenization & Embedding
LLMs
Large language models
Predict Result

Figure 1: The workflow of our analysis process. Our analysis workflow involves processing sequence data using different tokenization and embedding methods with various LLMs, such as GPTs and Gemini. To analyze the preferences of LLMs, we compute the seasonal and trend strength inside the datasets. Our experiments illuminate that LLMs prefer series with higher seasonal and trend strengths. To elucidate the rationale behind our findings, we demand the LLMs identify the underlying periods, revealing that the model can recognize the underlying periods in most cases. In addition, to improve the performance of time series forecasting, we propose two approaches to the user input: for the input prompt, we incorporate human knowledge regarding the dataset sources, and for the input sequence, we reprogram the data into natural language sequences. Both methods result in substantially improved model performance.

[13], we define the strength of the trend and the seasonality as follows:

$$Q_T = 1 - \frac{\mathrm{Var}(X_R)}{\mathrm{Var}(X_T + X_R)}, \quad Q_S = 1 - \frac{\mathrm{Var}(X_R)}{\mathrm{Var}(X_S + X_R)} \tag{5}$$

where $X_K \in R^K$, $X_S \in R^K$ and $X_R \in R^K$ denote the trend component, the seasonal component and the residual component respectively. The presented indices indicate the trend's strength and seasonality, providing a measure ranging up to 1. It is easy to find that a higher value indicates a stronger trend or seasonality within the time series. Throughout this paper, we use the word "higher strength" to represent the comparison of the strengths between different datasets. The assessment of strength is not based on a fixed level, as the concepts of "strong" and "weak" vary across different datasets and scenarios.

To further discern the LLMs' preferences for the specific segments of the input data, we add Gaussian noise to the original time series to create counterfactual examples. We start by defining a sliding window that constitutes 10% of the total length of the time series, and we set the sliding window to gradually move closer to the output sequence. This method allows us to assess the impact of different segments fairly and thereby infer the interpretability of the time series segments that LLMs predominantly focus on.

3.2 Preferences for Input Sequences

In this subsection, we investigate the input sequence preferences for time series forecasting with LLMs. We conduct experiments on real datasets with GPT-3.5-turbo and GPT-4-turbo, measuring model performance through MAPE. To further validate our findings, we also use GPT-3.5-turbo and Gemini-1.0-Pro to forecast multiple-period time series on synthesized datasets.

3.2.1 *Implementation Details*

Real Datasets: We conduct experiments on ten real-world datasets, including both those with clear patterns and those with irregular characteristics. The results are shown in Table 8. We apply the Seasonal-Trend decomposition using the LOESS (STL) technique [9] to decompose the original time series into trend, seasonal, and residual components. Subsequently, we compute the strengths of the trend strength Q_T and seasonal strength Q_S. To further understand the LLMs' preferences for the specific segments of the input data, we conduct the counterfactual analysis with a systematic permutation to the input time series. We first scale the sequence through max-min normalization. We then define a sliding window that constitutes 10% of the total length of the time series and add Gaussian noise into the data within this window data. Subsequently, the sliding window moves closer to the last known data point.

Synthesized Datasets: To further validate our findings and investigate the influence of the number of periods on model performance, we generate a dataset using the function $y = \alpha * x + \beta_1 * cos(2\pi f_1 * x) + \beta_2 * cos(2\pi f_2 * x) + \epsilon$. x ranges from 0 to 20 and ϵ follows the normal distribution $\mathcal{N}(0, 1)$.

3.2.2 *Key Findings*

Table 1: Correlation matrix between the strengths of the input time series and the model performance.

Metrics	GPT4-MAPE	GPT3.5-MAPE	Trend Strength Q_T	Seasonal Strength Q_S
GPT4-MAPE	1.00	0.99	-0.02	-0.68
GPT3.5-MAPE	0.99	1.00	-0.12	-0.67
Trend Strength Q_T	-0.02	-0.12	1.00	0.51
Seasonal Strength Q_S	-0.68	-0.67	0.51	1.00

After computing the Pearson correlation coefficients (PCC), we observe a nearly strong correlation between the strengths and model performance, showing that LLMs perform better when the input time series has a higher trend and seasonal strength, which is shown in Table 1. In the context of multi-period time series, the model performance worsens as the number of periods increases. It indicates that LLMs may have difficulty recognizing the multiple periods inherent in such datasets. Besides, for counterfactual analysis, as shown in Figure 2 and Figure 3, there is a noticeable increase in MAPE values when Gaussian noise is added to the latter segments, while the perturbation of the first part of the sequence has little effect on the prediction performance. Our findings reveal that LLMs are more sensitive to the end of input time series when forecasting. We show our full results in Figure 2 and Figure 3. As we move to the right along the x-axis, the closer it gets to the output sequence. It is also found that the initial part of the sequence has the least impact on the prediction accuracy. For the datasets with high seasonal strengths over 85%, such as WoolyDataset, and MonthlymilkDataset, more than 80% of the length of the time series has almost no effect on the model performance.

4. WHY DO LLMS FORECAST WELL ON DATA WITH HIGHER SEASONAL STRENGTHS?

Our findings show that LLMs demonstrate enhanced performance in time series forecasting with strong seasonal strengths. This raises the question: Why do LLMs perform well in forecasting datasets with marked seasonal patterns? To explore this phenomenon, we craft prompts that require LLMs to recognize the dataset's temporal pattern.

This approach is grounded in the hypothesis that LLMs are proficient in handling datasets with distinct seasonal attributes. By explicitly prompting LLMs to predict the dataset's period, we aim to leverage their inherent ability to discern and extrapolate from complex patterns, which sheds light on the mechanisms that underpin their superior performance in such contexts.

4.1 Implementation Details

To explore the phenomenon that LLMs forecast well on datasets with higher seasonal strengths, we design experiments to verify this phenomenon. We tokenize the input sequence and let the LLMs output the period directly. We use GPT-3.5-turbo, GPT-4-turbo, and Gemini-1.0-Pro to predict the periods. We have chosen five datasets with their seasonal strengths exceeding 85%. These datasets are readily available with clear seasonal patterns. In contrast, determining the specific periods of other irregular datasets is challenging, as they have no specific cycles. We record the predicted periods ten times and identify the mode period, which is the most frequently predicted value. We then compare the mode of these ten results with the real period. The mode is selected as the evaluation metric because, when considering the usage characteristics of LLMs, the output of this number best represents the model's normal performance. The results are shown in Table 3.

4.2 Key Findings

According to the results, we find that large language models can mostly determine the periodicity of a dataset. The true periods are determined here by the periodogram, which is commonly used to identify the dominant periods [14]. The multiples of the predicted period also align with the original data cycle. Consequently, we consider the prediction of these multiples to be accurate. We observe that LLMs generally perform well in predicting the period for most datasets with minimal fluctuations. Surprisingly, we discover that in the case of WoolyDataset and AusbeerDataset, which possess relatively short underlying periods, the predicted period is consistently 3 instead of the true period, 4. This discrepancy may be attributed to the LLMs' tendency to focus on cyclic patterns among individual digits rather than considering the entire sequence as a whole, a phenomenon that could also be interpreted as the model's identification of the underlying cycle. We leave a comprehensive analysis of this phenomenon in the future.

5. HOW TO LEVERAGE THESE INSIGHTS TO IMPROVE THE MODEL'S PERFORMANCE?

Based on the findings in the previous two sections, our focus is now on how to leverage these findings to further improve model performance. In this paper, we propose two approaches to the user input without additional fine-tuning: for the input prompt, we incorporate additional knowledge of the specific trend and seasonal patterns in the dataset, which gives the model a richer understanding of the underlying patterns. Regarding the input sequence, we transform the time series data into formats resembling natural language sequences rather than relying on the original tokenization. This approach leverages LLMs' superior capabilities with language sequences. Both methods achieve substantially improved model performance.

5.1 Dataset description and the External Knowledge incorporated in the Prompts

We briefly introduce the datasets we use, which also serve as the external knowledge incorporated into the prompts. Following [5], we downsample the input series to an hourly frequency, yielding a total of 267 observations and resulting in relatively small datasets. Additionally, we incorporate Memorization datasets published after September 2021, the cutoff date for GPT-3.5-turbo, to demonstrate the effectiveness of TimeLLM and our proposed methods. Finally, we implemented univariate time series forecasting to predict the 'OT' feature on the ETTh1 and ETTm2 datasets, focusing on the last 96 steps of the test set.

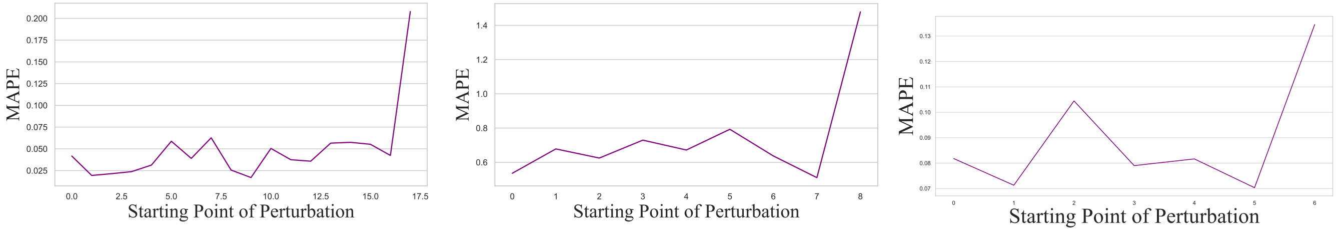
Table 2: Comparison test of traditional prediction methods.

AirPassengers		**AusBeer**	
Method	**MSE / MAE / MAPE**	**Method**	**MSE / MAE / MAPE**
Exponential Smoothing	2007.67 / 37.91 / 8.10	Exponential Smoothing	703.26 / 22.80 / 5.44
SARIMA	2320.47 / 39.80 / 8.46	SARIMA	**475.53** / 19.07 / 4.49
Cyclical Regression	2028.37 / 36.70 / 8.52	Cyclical Regression	989.31 / 26.29 / 6.13
AutoARIMA	8702.09 / 68.52 / 13.98	AutoARIMA	550.05 / 18.84 / 4.41
FFT	3274.46 / 46.38 / 10.59	FFT	7682.56 / 73.74 / 17.44
StatsForecastAutoARIMA	2952.52 / 45.41 / 9.71	StatsForecastAutoARIMA	559.46 / 20.56 / 4.86
Naive Mean	47703.65 / 204.25 / 44.61	Naive Mean	1885.72 / 30.66 / 6.68
Naive Seasonal	6032.80 / 62.87 / 14.18	Naive Seasonal	10828.02 / 96.35 / 23.39
Naive Drift	6505.79 / 72.21 / 17.50	Naive Drift	18507.61 / 128.23 / 30.91
Naive Moving Average	6032.80 / 62.87 / 14.18	Naive Moving Average	10828.02 / 96.35 / 23.39
N-Beats	3994.55 / 54.95 / 12.81	N-Beats	250.61 / 14.42 / 3.53
DeepAR	184222.64 / 421.99 / 98.42	DeepAR	16197.17 / 40.23 / 9.89
Prophet	7345.31 / 43.87 / 8.62	Prophet	6323.89 / 28.76 / 6.92
LLMTime with GPT-3.5-Turbo	6244.07 / 61.39 / 14.43	LLMTime with GPT-3.5-Turbo	841.68 / 23.59 / 5.62
LLMTime with GPT-4-Turbo	1317.9 / 55.49 / 11.18	LLMTime with GPT-4-Turbo	513.49 / 18.57 / 4.28
LLMTime with Gemini-1.0-pro	6392.21 / 63.57 / 14.03	LLMTime with Gemini-1.0-pro	397.78 / 14.36 / 3.27
LLMtime with Llama-2	1286.25 / 28.04 / 6.07	LLMtime with Llama-2	644.82 / 17.88 / 4.08

MonthlyMilk		**Sunspots**	
Method	**MSE / MAE / MAPE**	**Method**	**MSE / MAE / MAPE**
Exponential Smoothing	564.94 / 20.23 / 2.41	Exponential Smoothing	326750.49 / 499.78 / 3129.63
SARIMA	1289.76 / 32.78 / 3.87	SARIMA	2902.72 / 45.75 / 466.99
Cyclical Regression	3631.53 / 56.15 / 6.60	Cyclical Regression	3917.76 / 47.84 / 274.31
AutoARIMA	2682.67 / 42.82 / 5.20	AutoARIMA	4695.67 / 58.47 / 709.23
FFT	3453.96 / 45.62 / 5.48	FFT	3784.56 / 49.81 / 150.32
StatsForecastAutoARIMA	**186.14** / 10.64 / 1.28	StatsForecastAutoARIMA	8406.55 / 72.99 / 95.18
Naive Mean	19893.07 / 127.33 / 14.46	Naive Mean	4120.40 / 49.84 / 267.22
Naive Seasonal	4870.40 / 56.00 / 6.31	Naive Seasonal	4440.63 / 56.78 / 688.58
Naive Drift	3998.11 / 56.06 / 6.52	Naive Drift	5032.77 / 60.40 / 724.88
Naive Moving Average	4870.40 / 56.00 / 6.31	Naive Moving Average	4440.63 / 56.78 / 688.58
N-Beats	3140.89 / 51.57 / 6.07	N-Beats	4877.59 / 56.58 / 105.55
DeepAR	728289.50 / 851.30 / 99.22	DeepAR	3421.02 / 48.93 / 132.76
Prophet	663.41 / 25.76 / 2.92	Prophet	6303.57 / 76.83 / 67.97
LLMTime with GPT-3.5-Turbo	7507.13 / 66.28 / 112.77	LLMTime with GPT-3.5-Turbo	6556.55 / 58.95 / 217.94
LLMTime with GPT-4-Turbo	4442.18 / 50.75 / 172.82	LLMTime with GPT-4-Turbo	3374.70 / 41.87 / 321.11
LLMTime with Gemini-1.0-pro	628.98 / 17.01 / 1.99	LLMTime with Gemini-1.0-pro	626.03 / 14.94 / 1.73
LLMtime with Llama-2	3410.20 / 41.40 / 240.25	LLMtime with Llama-2	4467.67 / 48.95 / 91.79

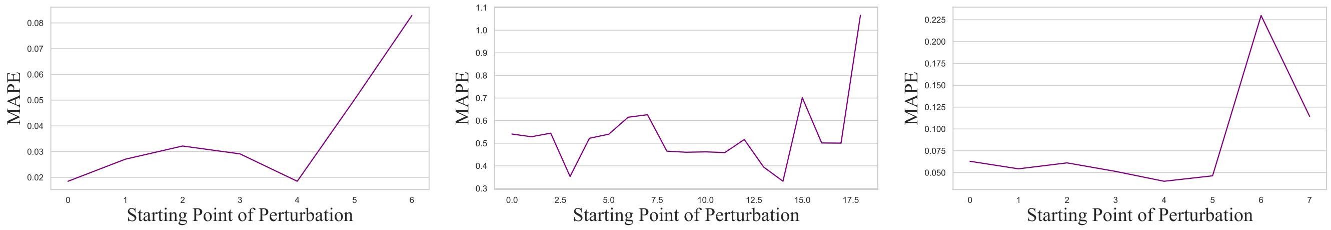
WineDataset		**WoolyDataset**	
Method	**MSE / MAE / MAPE**	**Method**	**MSE / MAE / MAPE**
Exponential Smoothing	23709576.52 / 3370.78 / 14.23	Exponential Smoothing	24925885.81 / 3548.19 / 14.98
SARIMA	1150166.94 / 966.57 / 20.76	SARIMA	812352.21 / 759.07 / 16.37
Cyclical Regression	7873785.27 / 2148.24 / 8.52	Cyclical Regression	1032574.82 / 962.72 / 22.14
AutoARIMA	698661.90 / 646.03 / 14.07	AutoARIMA	838852.91 / 786.25 / 16.84
FFT	1031170.45 / 867.83 / 18.60	FFT	1012255.35 / 945.20 / 20.80
StatsForecastAutoARIMA	20040877.37 / 2853.17 / 12.05	StatsForecastAutoARIMA	917617.19 / 858.57 / 18.91
Naive Mean	11557786.19 / 2200.04 / 8.80	Naive Mean	816762.31 / 764.73 / 16.12
Naive Seasonal	879447.22 / 724.23 / 15.52	Naive Seasonal	1051110.81 / 982.25 / 22.19
Naive Drift	9609576.04 / 1833.38 / 7.36	Naive Drift	812352.21 / 759.07 / 16.37
Naive Moving Average	9070696.99 / 1719.17 / 6.90	Naive Moving Average	1032574.82 / 962.72 / 22.14
N-Beats	5418377.00 / 1887.30 / 7.68	N-Beats	653104.31 / 743.54 / 15.96
DeepAR	715027008.00 / 26236.14 / 89.91	DeepAR	243831.14 / 4897.85 / 94.89
Prophet	4846922.27 / 2201.57 / 8.27	Prophet	365241.98 / 891.70 / 34.65
LLMTime with GPT-3.5-Turbo	30488.60 / 388.28 / 15.83	LLMTime with GPT-3.5-Turbo	526903.08 / 574.58 / 12.00
LLMTime with GPT-4-Turbo	22488.17 / 253.08 / 9.98	LLMTime with GPT-4-Turbo	942987.19 / 871.64 / 18.55
LLMTime with Gemini-1.0-pro	258584.78 / 3645.23 / 14.60	LLMTime with Gemini-1.0-pro	64.92 / 6.39 / 7.04
LLMtime with Llama-2	951194.94 / 240.08 / 9.45	LLMtime with Llama-2	675062.52 / 736.04 / 15.83

HeartRateDataset		**Weather**	
Method	**MSE / MAE / MAPE**	**Method**	**MSE / MAE / MAPE**
Exponential Smoothing	11.16 / 1.38 / 1.49	Exponential Smoothing	1684.38 / 31.60 / 6.79
SARIMA	12.98 / 1.34 / 1.61	SARIMA	1943.81 / 33.33 / 7.09
Cyclical Regression	13.58 / 1.31 / 1.20	Cyclical Regression	1700.73 / 30.77 / 7.15
AutoARIMA	13.26 / 1.25 / 1.39	AutoARIMA	7315.10 / 57.44 / 11.70
FFT	13.95 / 1.16 / 1.34	FFT	2752.02 / 38.90 / 8.87
StatsForecastAutoARIMA	10.53 / 1.27 / 1.39	StatsForecastAutoARIMA	2479.55 / 38.06 / 8.16
Naive Mean	12.02 / 1.27 / 1.26	Naive Mean	39879.84 / 168.27 / 36.44
Naive Seasonal	10.55 / 1.32 / 1.31	Naive Seasonal	5057.47 / 52.81 / 11.89
Naive Drift	10.60 / 1.15 / 1.30	Naive Drift	5466.23 / 60.58 / 14.70
Naive Moving Average	12.13 / 1.27 / 1.34	Naive Moving Average	5057.47 / 52.81 / 11.89
N-Beats	72.11 / 7.10 / 7.40	N-Beats	4532.84 / 39.21 / 23.49
DeepAR	286.82 / 15.67 / 16.36	DeepAR	6325.75 / 35.97 / 16.59
Prophet	88.93 / 10.97 / 6.54	Prophet	3768.15 / 29.36 / 24.01
LLMTime with GPT-3.5-Turbo	76.83 / 7.15 / 7.42	LLMTime with GPT-3.5-Turbo	224.54 / 3.07 / 0.83
LLMTime with GPT-4-Turbo	988.14 / 26.57 / 29.22	LLMTime with GPT-4-Turbo	111.65 / 2.40 / 0.64
LLMTime with Gemini-1.0-pro	57.96 / 6.01 / 6.66	LLMTime with Gemini-1.0-pro	176.32 / 3.72 / 0.75
LLMtime with Llama-2	75.58 / 7.11 / 7.94	LLMtime with Llama-2	215.39 / 4.07 / 1.31



(a) TSMCStock (b) IstanbulTrafficDataset (c) MonthlyMilk

Figure 2: Experiments of Sequence Focused Attention Through Counterfactual Explanation on GPT-3.5-turbo.



(a) TSMCStock (b) IstanbulTrafficDataset (c) MonthlyMilk

Figure 3: Experiments of Sequence Focused Attention Through Counterfactual Explanation on Gemini-Pro-1.0.

5.2 External Knowledge Enhancing Time Series Forecasting

We introduce a novel method to improve the performance of large language models for time series forecasting. The core idea of this part is to use the knowledge obtained from the pre-training stage to help predict. We provide the large language model with some basic information about the current dataset, such as the background of the data collection, and this process does not involve data leakage. We incorporate our tests on the data leakage in Section 5.5. We do not provide the LLMs with any statistical information, such as the periods or trends. This approach ensures that the LLMs forecast the time series entirely based on the data and their prior knowledge. Let V_s denote the initial prompt representing the original time sequence, and let z denote the additional information. Consequently, the new prompt V_e can be expressed as: $V_e = z + V_s$.

5.2.1 *Implementation Details*

We input the dataset's external knowledge through prompts before the sequence's input. The external knowledge of each dataset is presented in subsection 5.1. The results are shown in Table 8, where LLMTime Prediction refers to the approach described by [5] without any modifications.

5.2.2 *Key Findings*

As shown in Table 5, this method achieves improved performance in most scenarios. Besides, GPT-4-turbo generally performs better than GPT-3.5-turbo on MSE, MAE, and MAPE, especially on AirPassengers, AusBeer, and other datasets. Llama-2 significantly outperforms GPT-3.5-turbo and GPT-4-turbo in terms of MSE and MAE metrics on some datasets (e.g., Wooly, ETTh1, ETTm2), indicating that it can capture data features more accurately. Using External Knowledge Enhancing, Gemini outperforms other models on MonthlyMilk, Sunspots, Wooly, and HeartRate Datasets, but performs poorly on other datasets.

5.3 Natural Language Paraphrasing Time Series Forecasting

In this subsection, we conduct experiments on the natural language paraphrasing of the input time sequences. This strategy capitalizes on the advanced abilities of large language models in handling language sequences. It is motivated by the fact that LLMs are insensitive by the order of magnitude and size of digits [15].

We use natural language to describe the trend between consecutive values. For instance, given a time series X where $X = [X_1, X_2, X_3, \ldots, X_n]$, we describe the trend from X_t to X_{t+1} as follows: "The value rises from X_t to X_{t+1}, and falls from X_{t+1} to X_{t+2}...". The string we get here is our natural language paraphrasing sequence. After generating responses based on the string, we extract the values from the text and construct the predicted time series.

5.3.1 *Implementation Details*

We use GPT-3.5-Turbo, GPT-4-turbo, Llama-2 and Gemini-Pro-1.0 to forecast the time series, where part of the results are presented in Table 4 due to the page limit.

5.3.2 *Key Findings*

According to the results in Table 4, we find that enhancing LLM through natural language paraphrasing improves time series forecasting on most datasets. For instance, GPT-3.5-turbo and GPT-4-turbo perform better on most datasets, especially on Natural Language Paraphrasing methods. Gemini

Table 3: Results and comparison of time series period prediction based on GPT-3.5-turbo and Gemini.

Model	Dataset	Period										Real	Mode
GPT-3.5-turbo	**AirPassengersDataset**	24	24	7	24	12	24	11	24	24	24	12	24
	WineDataset	11	12	24	24	24	20	24	24	24	24	12	24
	MonthlyMilkDataset	6	9	12	9	12	12	12	12	12	11	12	12
	WoolyDataset	4	3	4	3	3	4	3	3	6	3	4	3
	AusBeerDataset	3	3	3	3	3	3	3	3	3	3	4	3
Gemini-Pro-1.0	**AirPassengersDataset**	11	12	12	4	12	12	12	12	12	12	12	12
	WineDataset	10	12	24	12	6	12	12	24	12	12	12	12
	MonthlyMilkDataset	16	12	12	12	12	39	12	11	12	12	12	12
	WoolyDataset	5	7	4	5	4	4	4	4	5	6	4	4
	AusBeerDataset	4	4	4	2	5	5	4	3	5	7	4	4
GPT-4-turbo	**AirPassengersDataset**	12	12	12	12	12	12	12	12	12	12	12	12
	WineDataset	7	6	6	6	7	7	6	6	6	6	12	6
	MonthlyMilkDataset	10	12	12	12	12	14	12	12	12	12	12	12
	WoolyDataset	5	5	7	5	5	5	7	5	5	4	4	5
	AusBeerDataset	4	4	4	4	6	4	4	4	6	4	4	4

outperforms other LLMs on Wooly and AusBeer datasets but underperforms on others with natural language paraphrasing. All these results demonstrate the superior performance of our methods.

5.4 Computational Cost

For reference, we list the average token length cost associated with external knowledge enhancement and natural language paraphrasing. Avg Token Length(ori) is the prompt Length of the unexecuted method, and Avg Token Length(EKE, NLP) is the prompt length after executing the corresponding policy. It is noted that Natural Language Paraphrasing is judged one by one through hard coding. Besides, there is a length check after transformation, so it is guaranteed that a certain length can be obtained each time. The results are shown in Table 7. Several key observations can be made: the original TimeLLM method maintains a uniform token length of 200 across all datasets, providing a stable baseline. EKE results in a slight increase in token length, ranging from 7% to 12%, suggesting a good balance between incorporating additional context and maintaining computational efficiency. In contrast, NLP leads to a more substantial increase in token numbers.

5.5 Tests on Data Leakage

We indirectly explore the data leakage problem by asking LLMs if they can identify the dataset name, the first 20 steps of the predicted dataset, and identify the dataset based on the first 20 steps of the time series data points. The results show that although GPT and Gemini can identify and determine data sets with limited information, they generally do not have detailed sequence data knowledge for a wider range of data sets Table 6.

6. RELATED WORK

In this section, we review two lines of research that are most relevant to ours.

6.1 Traditional Time Series Forecasting

Two commonly used traditional time series analysis methods are the ARIMA method [16] and the exponential smoothing method [17]. The ARIMA model is a classic forecasting method that breaks down a time series into auto-regressive (AR), difference (I), and moving average (MA) components to make predictions. On the other hand, exponential smoothing is a straightforward yet effective technique that forecasts future values by taking a weighted average of past observations. The ARIMA model requires testing data stationarity and selecting the right order. However, the exponential smoothing method is not affected by outliers; it is only suitable for stationary time series, and its accuracy in predicting future values is lower than that of the ARIMA model.

6.2 LLMs for Time Series Forecasting

The first family of methods involves either pre-training a foundational large language model or fine-tuning existing LLMs by leveraging extensive time-series data [6, 18–20]. For instance, [6] aimed to build the foundational models for time series and investigate its scaling behavior. [21] proposed a two-stage fine-tuning strategy for handling multivariate time-series forecasting. Although these studies contribute significantly to understanding foundational models, they require considerable computing resources and expertise in fine-tuning procedures. Moreover, the details of the model may not be disclosed for commercial purposes [18, 22], which impedes future research. Additionally, in scenarios with limited data available, there is insufficient information for training or fine-tuning.

In contrast, the second family of methods does not involve model parameter finetuning. These methods either create appropriate prompts or reprogramme inputs to effectively handle time series data [5, 7, 23, 24]. [7] tokenizes the time series and manages to embed those tokens, and [23] reprogrammed the time series data with text prototypes before feeding them to the LLMs. These studies illuminate the characteristics of time series data and devise methods to align them with LLMs. However, they lack an analysis of the ability and bias in forecasting time series. The most related work to us is [5], though it lacks a quantitative analysis of the preference for the time series in LLMs, and it fails to explore the impact of input forms and prompt contents, such as converting the numerical time series into the natural language sequences and incorporating the background information into the prompt. Our work fills the gap, and we expect our work to be the benchmark for time-series analysis and provide insights for subsequent research.

7. CONCLUSIONS AND FUTURE WORK

In this work, we investigate the key preferences of LLMs in the domain of time series forecasting under the zero-shot

Table 4: The results of natural language paraphrasing of sequences and baseline comparison(Partial).

Models	Datasets	Natural Language Paraphrasing			LLMTime Prediction		
		MSE	MAE	MAPE	MSE	MAE	MAPE
GPT-3.5-Turbo (GPT-3.5-turbo-1106)	AirPassengers	267.66	3.66	0.99	6244.07	61.39	14.43
	AusBeer	598.45	5.81	1.36	841.68	23.59	5.62
	GasRateCO2	3.16	0.46	0.85	10.88	2.66	4.73
	MonthlyMilk	968.69	8.61	1.02	7507.13	66.28	112.77
	Sunspots	251.61	4.27	20.42	6556.55	58.95	217.94
	HeartRate	4.38	0.55	0.57	76.83	7.15	7.42
	Istanbul-Traffic	224.17	3.74	8.81	335.05	6.75	11.68
	ETTh1	1.21	0.48	54.17	5.64	2.71	1.625
	ETTm2	0.81	0.36	27.33	3.46	2.17	1.178
GPT-4-Turbo (GPT-4-turbo-preview)	AirPassengers	133.10	2.87	0.80	1286.25	28.04	6.07
	AusBeer	661.80	7.24	1.63	513.49	18.57	4.28
	GasRateCO2	2.28	0.41	0.75	7.27	2.32	4.18
	MonthlyMilk	413.63	4.94	0.57	4442.18	50.75	172.82
	Sunspots	194.52	5.30	16.10	3374.70	41.87	321.11
	HeartRate	11.64	1.21	1.30	988.14	26.57	29.22
	Istanbul-Traffic	176.91	3.88	9.67	195.33	5.53	10.03
	ETTh1	1.20	0.49	47.62	4.73	1.53	3.282
	ETTm2	0.45	0.27	23.62	2.30	1.034	1.607
Llama-2 (llama-2-13B)	AirPassengers	751.34	6.77	1.53	1317.9	55.49	11.18
	AusBeer	591.75	23.25	5.41	644.82	17.88	4.08
	GasRateCO2	10.16	2.89	5.16	12.78	2.97	5.47
	MonthlyMilk	851.17	84.83	9.46	3410.20	41.40	240.25
	Sunspots	1483.29	33.27	17.79	4467.67	48.95	91.79
	HeartRate	49.8	5.84	6.53	75.58	7.11	7.94
	Istanbul-Traffic	306.80	5.39	7.24	438.28	7.28	9.81
	ETTh1	1.47	0.87	58.34	4.84	1.79	3.178
	ETTm2	0.84	0.41	29.86	3.31	2.07	2.153
Gemini-Pro-1.0 (gemini-1.0-pro)	AirPassengers	4474.54	31.54	7.02	6392.21	63.57	14.03
	AusBeer	278.45	10.05	2.29	397.78	14.36	3.27
	GasRateCO2	13.29	2.50	4.38	18.99	3.57	6.46
	MonthlyMilk	440.29	11.91	1.39	628.98	17.01	1.99
	Sunspots	438.29	10.47	1.21	626.03	14.94	1.73
	HeartRate	40.57	4.20	4.67	57.96	6.01	6.66
	Istanbul-Traffic	267.43	5.69	8.37	321.56	7.32	9.71
	ETTh1	1.17	0.74	54.86	4.84	1.79	3.178
	ETTm2	0.88	0.39	21.82	3.31	2.07	2.153

setting, revealing a proclivity for data with distinct trends and seasonal patterns. Through a blend of real and synthetic datasets, coupled with counterfactual experiments, we have demonstrated LLMs' improved forecasting performance with time series that exhibit clear periodicity. Besides, our results indicate that LLMs struggle with multi-period time series datasets, as they face difficulty in recognizing the distinct periods within them. Our findings also suggest that large language models are more sensitive to the segment of input sequences closer to the last known data than other locations. Lastly, experimental results indicate that our proposed strategies of incorporating external knowledge and transforming numerical sequences into natural language formats have yielded substantial improvements in accuracy.

8. REFERENCES

[1] C. Peng, X. Yang, A. Chen, K. E. Smith, N. PourNejatian, A. B. Costa, C. Martin, M. G. Flores, Y. Zhang, T. Magoc *et al.*, "A study of generative large language model for medical research and healthcare," *NPJ Digital Medicine*, 2023.

[2] C. Ledro, A. Nosella, and A. Vinelli, "Artificial intelligence in customer relationship management: literature review and future research directions," *Journal of Business & Industrial Marketing*, 2022.

[3] A. H. Huang, H. Wang, and Y. Yang, "Finbert: A large language model for extracting information from financial text," *Contemporary Accounting Research*, 2023.

[4] M. Jin, Q. Yu, H. Zhao, W. Hua, Y. Meng, Y. Zhang, M. Du *et al.*, "The impact of reasoning step length on large language models," *arXiv preprint arXiv:2401.04925*, 2024.

[5] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, "Large language models are zero-shot time series forecasters," *arXiv preprint arXiv:2310.07820*, 2023.

[6] K. Rasul, A. Ashok, A. R. Williams, A. Khorasani, G. Adamopoulos, R. Bhagwatkar, M. Biloš, H. Ghonia, N. V. Hassen, A. Schneider *et al.*, "Lag-llama: Towards foundation models for time series forecasting," *arXiv preprint arXiv:2310.08278*, 2023.

[7] C. Sun, Y. Li, H. Li, and S. Hong, "Test: Text prototype aligned embedding to activate llm's ability for time series," *arXiv preprint arXiv:2308.08241*, 2023.

[8] Hugging Face, "Chapter 6.5 of nlp course," 2023, accessed: 2023-02-10.

[9] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition," *J. Off. Stat*, 1990.

Table 5: The results of external knowledge enhancement and baseline comparison.

Models	Dataset	External Knowledge Enhancing			LLMTime Prediction		
		MSE	MAE	MAPE	MSE	MAE	MAPE
GPT-3.5-turbo-1106	**AirPassengers**	3713.99	50.37	10.88	6244.07	61.39	14.43
	AusBeer	669.01	21.82	5.12	841.68	23.59	5.62
	GasRateCO2	16.47	3.36	5.97	10.88	2.66	4.73
	MonthlyMilk	4781.26	55.45	6.25	7507.13	66.28	112.77
	Sunspots	7072.42	62.61	194.29	6556.55	58.95	217.94
	HeartRate	59.83	6.44	6.75	76.83	7.15	7.42
	Istanbul-Traffic	888.31	28.16	60.11	1321.44	48.7	7.47
	ETTh1	2.65	1.01	132.13	5.64	2.71	1.625
	ETTm2	2.00	0.89	201.84	3.46	2.17	1.178
GPT-4-turbo-preview	**AirPassengers**	1262.24	30.54	6.80	1286.25	28.04	6.07
	AusBeer	345.59	15.70	3.69	513.49	18.57	4.28
	GasRateCO2	6.99	2.29	4.21	7.27	2.32	4.18
	MonthlyMilk	2209.33	44.02	5.12	4442.18	50.75	172.82
	Sunspots	4571.92	50.24	334.30	3374.70	41.87	321.11
	HeartRate	78.99	6.96	7.90	988.14	26.57	29.22
	Istanbul-Traffic	954.88	26.92	47.29	1291.17	32.16	6.46
	ETTh1	2.70	1.06	129.99	4.73	1.53	3.282
	ETTm2	1.18	0.79	291.67	2.30	1.034	1.607
Llama-2	**AirPassengers**	3713.99	50.37	10.88	1286.25	28.04	6.07
	AusBeer	893.56	21.49	4.87	644.82	17.88	4.08
	GasRateCO2	11.38	3.04	5.49	12.78	2.97	5.47
	MonthlyMilk	4722.32	60.36	7.05	3410.20	41.40	240.25
	Sunspots	4000.19	46.45	138.69	4467.67	48.95	91.79
	HeartRate	112.17	7.86	8.93	75.58	7.11	7.94
	Istanbul-Traffic	979.15	26.70	45.57	1531.37	34.74	7.42
	ETTh1	4.15	1.65	408.11	4.84	1.79	3.178
	ETTm2	3.08	1.47	810.56	3.31	2.07	2.153
Gemini-1.0-pro	**AirPassengers**	5237.85	51.92	11.08	6392.21	63.57	14.03
	AusBeer	325.45	10.84	1.86	397.78	14.36	3.27
	GasRateCO2	15.54	3.23	4.43	18.99	3.57	6.46
	MonthlyMilk	491.26	15.18	1.13	628.98	17.01	1.99
	Sunspots	491.64	11.15	1.27	626.03	14.94	1.73
	HeartRate	47.45	4.83	4.67	57.96	6.01	6.66
	Istanbul-Traffic	1253.74	28.25	5.42	1531.37	34.74	7.42
	ETTh1	2.92	1.45	2.88	4.84	1.79	3.178
	ETTm2	2.00	1.74	1.22	3.31	2.07	2.153

Table 6: Summary of tests on different datasets.

Datasets	**Acknowledge Test (GPT)**	**Acknowledge Test (Gemini)**	**Series Test (GPT)**	**Series Test (Gemini)**	**Dataset Detection (GPT)**	**Dataset Detection (Gemini)**
AirPassengers	Yes	Yes	Yes	No	No	No
AusBeer	No	Yes	No	No	No	No
GasRateCO2	No	Yes	No	No	No	No
MonthlyMilk	Yes	Yes	No	No	No	No
Sunspots	Yes	Yes	No	No	No	No
Wine	Yes	Yes	No	No	No	No
Wooly	No	No	No	No	No	No
HeartRate	Yes	Yes	No	No	No	No

[10] J. Herzen, F. Lässig, S. G. Piazzetta, T. Neuer, L. Tafti, G. Raille, T. Van Pottelbergh, M. Pasieka, A. Skrodzki, N. Huguenin *et al.*, "Darts: User-friendly modern machine learning for time series," *Journal of Machine Learning Research*, vol. 23, no. 124, pp. 1–6, 2022.

[11] R. W. Godahewa, C. Bergmeir, G. Webb, R. Hyndman, and P. Montero-Manso, "Monash time series forecasting archive," in *NeurIPS Systems Track on Datasets and Benchmarks*, 2021.

[12] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *AAAI 2021*, 2021.

[13] X. Wang, K. Smith, and R. Hyndman, "Characteristic-based clustering for time series data," *Data mining and knowledge Discovery*, 2006.

[14] M. S. Bartlett, "Periodogram analysis and continuous spectra," *Biometrika*, 1950.

Table 7: Comparison of Avg Token Lengths among Original TimeLLM method, External Knowledge Enhancing and Natural Language Paraphrasing.

Datasets	Avg Token Length (ori)	Avg Token Length (EKE)	Avg Token Length (NLP)
AirPassengers	200	224	797
AusBeer	200	220	797
GasRateCO2	200	211	797
MonthlyMilk	200	218	797
Sunspots	200	217	797
Wine	200	217	797
Wooly	200	216	797
HeartRate	200	214	797

Table 8: Model performance in the analysis of LLMs' preferences.

Dataset Name	GPT4-MAPE	GPT3.5-MAPE	Trend Strength	Seasonal Strength
AirPassengersDataset	6.80	9.98	1.00	0.98
AusBeerDataset	3.69	5.12	0.99	0.96
MonthlyMilkDataset	5.12	6.25	1.00	0.99
SunspotsDataset	334.30	194.29	0.81	0.28
WineDataset	10.90	14.98	0.67	0.92
WoolyDataset	20.41	19.26	0.96	0.82
IstanbulTrafficGPT	47.29	60.11	0.31	0.72
GasRateCO2Dataset	4.21	5.97	0.65	0.50
HeartRateDataset	7.90	6.75	0.42	0.49
TurkeyPower	3.36	3.52	0.90	0.88

[15] R. Shah, V. Marupudi, R. Koenen, K. Bhardwaj, and S. Varma, "Numeric magnitude comparison effects in large language models," in *Findings of ACL 2023*, 2023.

[16] G. E. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *Journal of the American statistical Association*, vol. 65, no. 332, pp. 1509–1526, 1970.

[17] E. S. Gardner Jr, "Exponential smoothing: The state of the art—part ii," *International journal of forecasting*, vol. 22, no. 4, pp. 637–666, 2006.

[18] A. Garza and M. Mergenthaler-Canseco, "Timegpt-1," *arXiv preprint arXiv:2310.03589*, 2023.

[19] A. Das, W. Kong, R. Sen, and Y. Zhou, "A decoder-only foundation model for time-series forecasting," *arXiv preprint arXiv:2310.10688*, 2023.

[20] D. Cao, F. Jia, S. O. Arik, T. Pfister, Y. Zheng, W. Ye, and Y. Liu, "Tempo: Prompt-based generative pre-trained transformer for time series forecasting," *arXiv preprint arXiv:2310.04948*, 2023.

[21] C. Chang, W.-Y. Wang, W.-C. Peng, and T.-F. Chen, "Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters," 2024.

[22] C. Zhang, M. Jin, Q. Yu, C. Liu, H. Xue, and X. Jin, "Goal-guided generative prompt injection attack on large language models," *arXiv preprint arXiv:2404.07234*, 2024.

[23] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan *et al.*, "Time-llm: Time series forecasting by reprogramming large language models," *arXiv preprint arXiv:2310.01728*, 2023.

[24] H. Xue and F. D. Salim, "Promptcast: A new prompt-based learning paradigm for time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, 2023.