Unifying Knowledge in Agentic LLMs: Concepts, Methods, and Recent Advancements

Lihui Liu
lihuil2@illinois.edu
Wayne State University
Detroit, Michigan, USA

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

Abstract

Large language models have demonstrated remarkable capabilities in text generation and problem solving, yet they continue to face fundamental challenges such as hallucination, lack of factual grounding, and limited reasoning reliability. The core of these issues lies the question of how LLMs acquire and use *knowledge*. While internal knowledge embedded in model parameters enables impressive generalization, it is often insufficient for up-to-date or domain-specific tasks. External knowledge integration, such as retrieval-augmented generation (RAG), provides grounding and factuality but introduces challenges of retrieval quality, latency, and reliability. Beyond these paradigms, recent advances in *agentic LLMs* extend models from passive generators to active problem solvers that can reason, plan, and interact with external tools. This survey provides a unified, knowledge-centric perspective on LLMs, organized along three complementary dimensions: (i) reactive: internal knowledge, (ii) lightly-active: external knowledge, and (iii) proactive: agentic knowledge utilization for reasoning and tool interaction. We provide a taxonomy of knowledge usage in LLMs, analyze their respective strengths and limitations, and highlight how these paradigms interact in real-world systems. Finally, we identify open challenges to facilitate future research.

CCS Concepts

• **Computing methodologies** → **Reasoning about belief and knowledge**; • **Information systems** → **Data mining**.

Keywords

Knowledge graph reasoning, neural symbolic reasoning, knowledge graph question answering

1 Introduction

Knowledge lies at the core of how large language models (LLMs) reason [15, 47, 52], generate responses [1, 45], and support decision-making [3, 53]. The billions of parameters in modern LLMs encode an impressive amount of internalized world knowledge acquired during large-scale pretraining. This internal knowledge enables LLMs to perform diverse tasks with few or even zero examples. However, such parametric knowledge is often incomplete, outdated, or unreliable, particularly when applied to specialized or dynamic domains. As a result, relying solely on the internal memory of an LLM limits its factual accuracy, interpretability, and trustworthiness when answering questions.

To overcome these limitations, recent research has sought to augment LLMs with external sources of knowledge—ranging from large unstructured corpora to structured databases and knowledge graphs—so that generated outputs can be grounded in verifiable evidence [12, 19, 40]. This line of work, broadly referred to as retrieval-augmented generation (RAG), has shown that explicit access to external information can substantially improve factuality, reasoning, and adaptability across domains. In parallel, the emergence of agentic LLMs [3, 26, 51, 53], which can autonomously plan, reason, and act through interaction with tools and environments, represents a new paradigm of knowledge use in action. Rather than passively generating text, such models leverage both internal and external knowledge to make decisions, execute plans, and continuously update their understanding through feedback and exploration.

Together, these developments reveal that knowledge is not merely a static component encoded in parameters but a dynamic process—one that governs how LLMs organize, access, and apply information to reason and act effectively. Yet, despite the growing body of research, existing studies tend to examine these directions in isolation. For instance, work on in-context learning (ICL) [4, 11, 47] primarily explores how models exploit internal knowledge through prompting and contextual adaptation. In contrast, retrieval-augmented approaches [12, 19, 40] focus on grounding model outputs with external evidence. More recent efforts on LLM-based agents [3, 27, 51] investigate tool use, planning, and interaction, but often from a systems or application-oriented perspective rather than from a unifying theory of knowledge utilization.

What remains missing in the current literature is a unifying conceptual framework that bridges these perspectives and offers a holistic understanding of how knowledge is represented, retrieved, and applied across different paradigms of large language model reasoning. While prior surveys have reviewed individual aspects—such as prompting and in-context learning, retrieval-augmented generation, or LLM-based agents—there is still no comprehensive synthesis that explains how these approaches collectively contribute to the evolving landscape of knowledge-centric intelligence.

This survey aims to fill that gap by proposing a knowledge-centric taxonomy that unifies three complementary modes of knowledge use in LLMs, ranging from internal recall to active reasoning and autonomous interaction (see Fig. 3). (1) Reactive use of internal knowledge — LLMs rely on their parametric memory and contextual reasoning to solve tasks based solely on internalized knowledge. (2) Lightly-active use of external knowledge — LLMs ground their generation in verifiable external sources via retrieval, database queries, or structured representations such as knowledge graphs. (3) Proactive use of knowledge in an agentic manner — LLMs integrate internal and external knowledge to plan, act, and adapt dynamically through interaction with the environment or external tools.

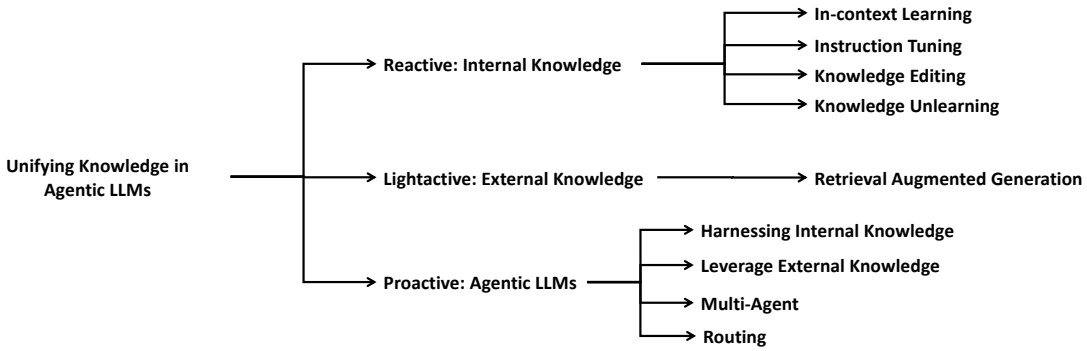

Figure 1: Overview of the survey: three paradigms of knowledge use in large language models.

Together, these three paradigms capture a continuum of knowledge utilization, ranging from passive recall to grounded reasoning and ultimately to autonomous, knowledge-driven behavior. This perspective illustrates how the role of knowledge in LLMs has evolved—from a static asset stored in model parameters to an active process that dynamically shapes perception, reasoning, and action.

Building upon this taxonomy, our survey systematically reviews recent advances across these three dimensions, discussing how knowledge is acquired during pretraining, dynamically retrieved or updated, and utilized for reasoning and planning in interactive environments. We further examine the interplay among these paradigms, identifying shared mechanisms such as memory management, reasoning chains, and feedback loops that unify them.

By organizing the literature around these three pillars, this survey provides a comprehensive and coherent understanding of how knowledge underpins the reasoning and decision-making capabilities of large language models. It also highlights key challenges and future directions for developing more reliable, explainable, and knowledge-driven AI systems that move beyond static text generation toward dynamic and trustworthy reasoning.

2 Foundations of Knowledge in LLMs

2.1 Background

Large Language Models (LLMs) are powerful deep neural networks built upon the Transformer architecture and trained on massive text corpora to model the probability distribution of natural language. Representative models such as GPT [1], PaLM [5], and LLaMA [45] are typically trained using an autoregressive objective, where the model learns to predict the next token in a sequence given its preceding context:

$$\max \sum_{t} \log P(x_t \mid x_{<t}),$$

with x_t representing the current token and $x_{<t}$ denoting all tokens before it. This self-supervised pretraining enables the model to implicitly learn grammar, semantics, world knowledge, and basic reasoning patterns from large-scale unlabeled data.

After pretraining, LLMs are typically adapted to downstream tasks through *supervised fine-tuning* [37]. In this stage, the model is trained on human-annotated input-output pairs (x, y), where x is a task-specific instruction or prompt, and $y = (y_1, \ldots, y_L)$ is the corresponding desired response. The model is optimized to maximize the conditional likelihood of the output sequence given the input:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{t=1}^{L} \log P_\theta(y_t \mid x, y_{<t}),$$

where $P_\theta(y_t \mid x, y_{<t})$ is the probability of generating token y_t based on the input and previous output tokens, and θ denotes the model parameters. This step helps the model follow instructions and perform specific tasks more reliably.

To further align model behavior with human preferences, values, and safety goals, LLMs are often refined via *Reinforcement Learning from Human Feedback (RLHF)* [37]. The RLHF pipeline begins with the collection of human preference data, where annotators rank several model-generated outputs for the same prompt. These rankings are used to train a *reward model* $r_\phi(x, y)$, which estimates the quality of a response y given a prompt x. Finally, the base model is fine-tuned using a reinforcement learning algorithm, commonly *Proximal Policy Optimization (PPO)* [39], to maximize the expected reward under the learned reward model:

$$\mathcal{L}_{\text{RLHF}}(\theta) = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \left[r_\phi(x, y) \right],$$

where π_θ is the current policy (the language model), and ϕ represents the parameters of the reward model. PPO ensures stable optimization by penalizing large deviations from the original policy during updates.

Together, these stages—pretraining, supervised fine-tuning, and RLHF—form a standard training framework that enables LLMs to exhibit strong language understanding, instruction-following behavior, and alignment with human intent. Despite their generative capabilities, LLMs still face challenges in factual consistency, verifiable reasoning, and robustness, as their outputs are shaped by statistical patterns in data rather than explicit logical or grounded inference mechanisms.

2.2 Internal vs. External Knowledge

Large language models (LLMs) can acquire and utilize knowledge through two complementary channels: *internal knowledge* and *external knowledge*. Internal knowledge [11, 47, 54] refers to the information implicitly encoded within the model parameters during large-scale pretraining. This enables models to recall facts, linguistic structures, and common-sense reasoning without explicit access

to external resources. By contrast, external knowledge [12, 19, 40] refers to information retrieved or accessed from outside sources at inference time, such as unstructured corpora, structured knowledge bases, or multimodal databases. External knowledge provides grounding, verifiability, and adaptability, especially in dynamic or specialized domains where internal memory may be insufficient or outdated.

2.3 A Taxonomy of Knowledge Usage

We propose a compact taxonomy that categorizes how large language models (LLMs) use knowledge into three complementary paradigms: *(1) internal knowledge*, *(2) external knowledge*, and *(3) agentic knowledge use*. This taxonomy captures both *where* information resides—within or outside the model's parameters—and *how* it is operationalized, ranging from static recall to dynamic reasoning and autonomous interaction.

(1) Internal knowledge refers to the information encoded or accessed without invoking any external sources. It encompasses the model's parametric memory and contextual reasoning abilities, which arise from large-scale pretraining. Several mechanisms fall under this category: (a) *In-context learning (ICL)* enables the model to adapt its behavior from examples or demonstrations provided directly in the prompt at inference time, functioning as a form of temporary contextual learning. (b) *Instruction tuning* aligns model behavior with human intent by fine-tuning on large collections of instruction–response pairs, effectively improving generalization across tasks. (c) *Knowledge editing* and *unlearning* techniques further refine internal knowledge by locally modifying or removing specific facts in the model parameters to enhance factual consistency, correct errors, or comply with privacy constraints. Together, these mechanisms illustrate how internal representations enable LLMs to reason reactively and flexibly using their intrinsic knowledge base.

(2) External knowledge encompasses methods that augment or ground LLM outputs with verifiable information residing outside their parameters. The most representative paradigm here is (a) *retrieval-augmented generation (RAG)*, in which the model retrieves relevant documents, knowledge-graph triples, or database entries at inference time and conditions its response on the retrieved evidence. (b) In addition to retrieval, emerging approaches explore direct *database querying* and integration with structured symbolic stores, allowing the model to reason over factual and relational information explicitly. (c) Some systems further employ *hybrid retriever–generator architectures*, where retrieval and generation are co-trained or iteratively refined for tighter coupling between knowledge access and use. Overall, these methods mitigate the limitations of parametric memory—such as incompleteness or temporal drift—by providing factual grounding and interpretability while maintaining generative flexibility.

(3) Agentic knowledge use represents the most dynamic form of knowledge utilization, in which LLMs function as autonomous decision-making entities that plan, reason, and act. Within this paradigm, (a) models *leverage internal reasoning*—such as chain-of-thought or instruction-tuned behaviors—as a substrate for high-level planning and reflection; (b) they *invoke external tools, APIs, or retrieval modules* (RAG is a special case of tool use) on demand to supplement missing knowledge or verify hypotheses; and (c) they often engage in *multi-agent collaboration*, coordinating with specialized agents such as retrievers, verifiers, or planners to decompose and solve complex tasks. Some advanced systems further support *dynamic routing*, where the model decides which knowledge source or agent to consult at each stage of reasoning. This paradigm embodies proactive knowledge use, integrating perception, reasoning, and action into an adaptive loop that extends beyond static text generation.

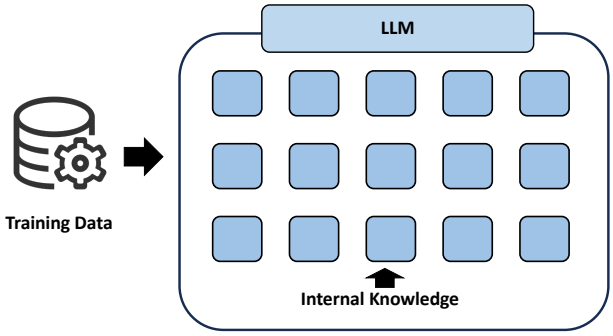

Figure 2: Internal Knowledge.

Overall, this three-part taxonomy—covering internal, external, and agentic paradigms—offers a unified lens for understanding how LLMs represent, access, and apply knowledge. It highlights the continuum from passive recall to autonomous reasoning, emphasizing the trade-offs between parametric memory, retrieval-based grounding, and agentic control. By articulating these categories and their mechanisms, this framework provides conceptual clarity for analyzing existing methods and guiding future research toward more reliable, explainable, and knowledge-driven LLM systems.

3 Internal Knowledge: Reactive

3.1 In-Context Learning

In-Context Learning (ICL) is one of the most distinctive capabilities of large language models (LLMs), providing a way to dynamically use and integrate knowledge. Rather than updating model parameters, ICL allows models to adapt to new tasks at inference time by conditioning on examples, instructions, or reasoning traces embedded directly in the input [11]. This approach leverages the knowledge already encoded in pretrained parameters and enriches it with knowledge presented in the immediate context, effectively blending static and contextual information in a single reasoning process. ICL highlights how LLMs can exploit knowledge in flexible ways. Prompting strategies, few-shot demonstrations, and structured reasoning traces such as chain-of-thought [47] exemplars serve as vehicles for injecting relevant knowledge into the model's decision process. These techniques show how users can curate and deliver domain-specific or task-specific knowledge on the fly, without retraining.

This paradigm offers several advantages from a knowledge perspective. By using natural language instructions (zero-shot) or a handful of examples (few-shot), ICL allows LLMs to quickly generalize to new domains and make use of unfamiliar knowledge. Its adaptability makes it possible to integrate knowledge that was

absent or only partially represented during pretraining, enabling rapid deployment across diverse application areas.

However, ICL has important limitations in managing and grounding knowledge. The finite context window restricts how much knowledge can be provided at once, which limits performance on knowledge-intensive tasks. Furthermore, because ICL does not explicitly verify or ground its outputs, it often generates hallucinations—fabricated information presented as fact. The absence of traceability means that the knowledge behind a prediction is opaque, reducing transparency and reliability. These issues highlight the need to extend ICL with external knowledge mechanisms, such as retrieval-augmented generation, to improve factuality and verifiability.

3.2 Instruction Tuning

Instruction Tuning is another central paradigm for adapting large language models (LLMs) to follow user-specified tasks more faithfully. Unlike In-Context Learning (ICL), which operates entirely at inference time, instruction tuning fine-tunes pretrained LLMs on a collection of curated datasets consisting of input-output pairs framed as natural language instructions [54]. Through this process, the model internalizes the mapping between instructions and desired behaviors, improving its ability to generalize to unseen tasks that share similar formats or intent. In effect, instruction tuning modifies the parameters of the LLM so that following instructions becomes an intrinsic behavior rather than an emergent property of prompting. Well-known examples include T5 [38], FLAN [46], and InstructGPT [37], which demonstrate that instruction tuning significantly enhances a model's usability by making it more responsive to natural instructions.

The benefits of instruction tuning are substantial. It improves robustness across domains, reduces the reliance on carefully engineered prompts, and provides a systematic approach to aligning LLM behavior with human preferences. Moreover, instruction-tuned models are more capable of handling task variation with minimal additional examples, narrowing the gap between artificial and human-like adaptability.

3.3 Knowledge Editing

Knowledge editing focuses on updating or modifying specific pieces of factual knowledge stored within large language models without retraining them from scratch. Unlike instruction tuning, which globally reshapes model behavior across tasks, knowledge editing seeks localized interventions: changing how the model responds to queries involving particular facts, entities, or relations, while preserving its overall performance and previously acquired knowledge [7, 32]. This makes knowledge editing especially valuable for time-sensitive or domain-specific updates, such as correcting outdated biomedical facts or incorporating new geopolitical events.

Formally, given a pretrained model f_θ, an editing algorithm aims to produce updated parameters $f_{\theta'}$ such that for a target query x^*, the output $f_{\theta'}(x^*)$ reflects the revised knowledge y^*. At the same time, for non-target queries $x \notin \mathcal{X}^*$, the outputs should remain close to their original predictions, minimizing unintended side effects. Many methods instantiate this principle through optimization

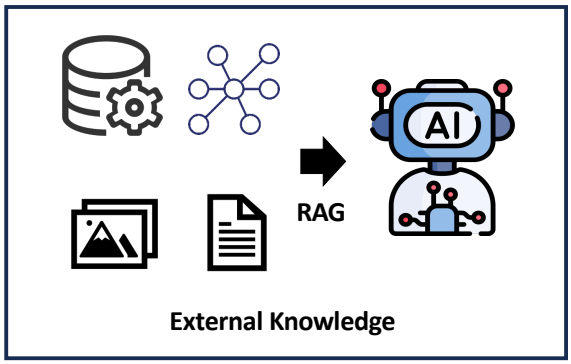

Figure 3: External Knowledge.

objectives of the form:

$$\min_{\Delta\theta} \mathcal{L}(f_{\theta+\Delta\theta}(x^*), y^*) + \lambda \mathbb{E}_{x \notin \mathcal{X}^*} [\mathrm{Dist}(f_{\theta+\Delta\theta}(x), f_\theta(x))] \quad (1)$$

where the first term enforces correctness of the edit and the second penalizes deviation from the model's prior knowledge.

In practice, knowledge editing ranges from parameter-based updates (e.g., ROME [32], MEMIT [33]) to interventions on hidden representations, balancing precision in updating target facts with generalization to diverse contexts. Nevertheless, edits can propagate undesired changes, fail to generalize beyond surface-level rephrasings, or struggle with multi-hop knowledge, motivating research into more robust, interpretable, and hybrid approaches that combine editing with retrieval or external knowledge for verifiable grounding.

3.4 LLM Knowledge Unlearning

Knowledge unlearning in LLMs aims to remove undesired or outdated knowledge while preserving unrelated information, which is critical for privacy, harm mitigation, and maintaining factual consistency in evolving KGs.

Model-based approaches typically modify parameters to erase specific knowledge, such as using Gradient Ascent (GA) [34] to invert gradients of undesired facts or variants that stabilize optimization via relabeled data. While effective at targeted removal, these methods can degrade retained knowledge and struggle to balance forgetting with preservation, especially given the interdependencies among entities and relations.

Evaluating unlearning remains challenging. Benchmarks like WHP [13], TOFU [30], and WMDP [20] measure fact removal using token-level or entity-level metrics, but most treat knowledge as independent and overlook relational structure. Multi-fact interactions have been explored, though current methods often rely on deterministic or rule-based evaluation, limiting scalability.

In KG–LLMs, unlearning is particularly delicate: removing one fact can inadvertently disrupt reasoning over related entities. Future work may focus on graph-aware algorithms, structural evaluation, and explainable methods to erase knowledge without compromising overall model reasoning and consistency.

4 External Knowledge: Lightly-active

While in-context learning (ICL) enables LLMs to exploit their internal parametric knowledge, it often suffers from limitations such

as hallucination, factual errors, and lack of verifiability. Retrieval-Augmented Generation (RAG) [12, 19, 40] has emerged as a complementary paradigm designed to address these issues. The central motivation behind RAG is to ground language model outputs in verifiable external sources, thereby improving factuality, reducing hallucinations, and providing transparency into the generation process. By incorporating retrieval mechanisms at inference time, RAG systems ensure that models are not solely dependent on internalized information, but can dynamically access up-to-date and domain-specific knowledge.

RAG systems are typically organized around several architectural patterns. The *classic pipeline* [40] follows a two-stage process: a retriever identifies relevant documents from an external corpus, which are then passed to the LLM for generation. More modular variants [12] separate retrieval and generation more explicitly, allowing fine-grained control over the knowledge integration process. Recent advances [31] also include *end-to-end retrieval-augmented training*, where both retrieval and generation components are optimized jointly, enabling the system to learn to retrieve the most useful context for the task at hand.

The effectiveness of RAG depends on the availability and quality of external knowledge sources. Commonly used sources include large-scale unstructured text corpora [8, 9, 14, 36, 44], structured repositories such as knowledge graphs [21–25, 28], and multimodal databases that integrate text, images, or other data modalities. Text corpora provide breadth and coverage, knowledge graphs offer structured and interpretable representations, and multimodal databases extend LLMs' reasoning beyond text alone. The choice of source often depends on the application, with hybrid approaches combining multiple knowledge types to improve robustness.

Research in RAG has advanced in several directions. Adaptive retrieval methods dynamically select the most relevant content based on context, reducing noise and improving precision. Multi-hop retrieval [43] enables the chaining of retrieval steps, supporting more complex reasoning across multiple documents. Another trend involves mixture-of-expert retrievers [16], where different retrieval modules specialize in distinct domains or modalities, and the system learns to route queries adaptively. Together, these advances push RAG beyond simple document lookup toward more sophisticated, context-aware grounding strategies.

Despite its promise, RAG faces several challenges. Retrieval quality remains a critical bottleneck, as noisy or irrelevant documents can mislead the generator [6]. Latency is another issue, as retrieval introduces additional computation that may hinder real-time applications [17]. Finally, ensuring reliable grounding is non-trivial: models may ignore retrieved evidence or selectively use it in ways that do not guarantee factual correctness [18]. Addressing these challenges is essential to make RAG both scalable and trustworthy.

5 Agentic LLMs: Proactive

Reactive vs Proactive: Traditional paradigms for leveraging knowledge in large language models, such as in-context learning and retrieval-augmented generation, primarily focus on how models passively access and integrate information. While effective in many scenarios, these approaches treat the model largely as a reactive system: it generates outputs based on prompts or retrieved context without actively initiating exploration, planning, or intervention. Agentic LLMs, in contrast, adopt a *proactive* stance. Rather than waiting for explicit queries, these models can autonomously identify relevant knowledge, anticipate information gaps, and plan multi-step actions to achieve objectives. This proactive behavior enables LLMs to actively operationalize knowledge, bridging the gap between static information retrieval and dynamic problem-solving. By reasoning about potential outcomes and taking initiative, agentic LLMs can efficiently navigate complex tasks, integrate diverse sources of knowledge, and adapt to changing environments.

Furthermore, proactive agentic behavior allows LLMs to better utilize knowledge in both parametric and non-parametric forms. Internally stored knowledge in model parameters can be applied strategically for reasoning and planning, while external knowledge sources—such as databases, APIs, or knowledge graphs—can be selectively queried to fill information gaps or verify hypotheses. This combination enhances both the effectiveness and reliability of the model, allowing it to act as an autonomous knowledge processor rather than a passive text generator. By enabling models to reason, plan, act, and interact with external tools or environments, agentic LLMs extend the scope of what AI systems can achieve. They are capable of sequential decision-making, iterative refinement, and adaptive behavior, all of which are crucial for real-world applications that demand more than single-step responses. In this sense, agentic LLMs represent a natural evolution from ICL and RAG toward models that can operationalize knowledge in a goal-directed and context-aware manner.

5.1 Harnessing Internal Knowledge

Agentic LLMs rely heavily on internal knowledge stored in their parameters, memory mechanisms, and reasoning capabilities:
Memory and Profile: In LLM agents, memory refers to the ability to retain and utilize past interactions, contextual information, and long-term knowledge about users or tasks, while profile represents structured information that defines the agent's role, attributes, preferences, and operational competencies. Together, they enable the agent to maintain continuity across sessions, adapt to user-specific needs, and perform domain-specific functions more effectively. Since both memory and profile are stored, organized, and accessed internally by the LLM (rather than retrieved externally), they are considered part of the LLM's internal knowledge, shaping how it reasons, plans, and interacts in dynamic environments.
Reasoning: In LLM agents, reasoning refers to the process of drawing logical inferences and making consistent conclusions based on available information, whether from the prompt, prior memory, or the model's internal knowledge. It allows the agent to connect facts, resolve ambiguities, and justify decisions beyond surface-level pattern matching. Since this ability emerges from the model's internal representations and learned structures—rather than depending on external retrieval—it is considered part of the LLM's internal knowledge.
Planning: In LLM agents, planning refers to the ability to generate and organize a sequence of coherent actions or reasoning steps that lead from the current state to a desired goal. Unlike simple response generation, planning enables the agent to anticipate future

requirements, break down complex tasks into manageable subtasks, and adapt strategies based on constraints or feedback. As this process relies on the model's internal reasoning capabilities—using its knowledge and learned patterns to decide how to act rather than retrieving instructions from outside—it is regarded as part of the LLM's internal knowledge and a key component of goal-directed autonomous behavior.

Frameworks like *ReAct* [53] interleave reasoning and planning, generating intermediate reasoning traces while leveraging internal knowledge for decision-making. *Reflexion* [41] adds feedback loops, enabling agents to evaluate and refine strategies based on past experiences.

5.2 Leverage External Knowledge

Agentic behavior in large language models can be substantially enhanced by incorporating external sources of knowledge. This integration can be broadly categorized into three complementary forms. (1) *Tool use*: Agents can invoke APIs, search engines, or specialized software interfaces to obtain up-to-date or task-specific information that is not encoded in their parametric memory. (2) *Knowledge graphs and databases*: Structured repositories provide factual grounding, enabling models to access verified relationships and reduce the risk of hallucination. (3) *Environment interaction*: Agents can act within simulated or real-world contexts to execute tasks, gather evidence, or refine their understanding through feedback.

Recent systems such as *AutoGPT* [51] demonstrate how these capabilities can be orchestrated within automated planning pipelines. By performing multi-step reasoning, tool invocation, and web-based information retrieval, such models achieve complex objectives. More generally, grounding agentic actions through retrieval-augmented generation (RAG) ensures that LLM behavior remains both factually accurate and temporally relevant while retaining open-ended reasoning abilities.

5.3 Multi-agent Collaboration

Agentic LLMs can also operate in multi-agent settings, where multiple models coordinate, negotiate, and share knowledge to accomplish complex objectives. Let $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$ denote a set of agents, each with internal state $\mathbf{s}_t^i$ and access to a subset of knowledge $\mathcal{K}_i$. Multi-agent frameworks enable several key capabilities.

(1) *Task decomposition*: Complex problems can be divided into sub-tasks and allocated to specialized agents, allowing each agent to focus on its strengths. This division of labor enhances efficiency and makes large-scale objectives more tractable.

(2) *Knowledge sharing*: Agents exchange intermediate results, reasoning traces, or learned insights to construct a richer shared context. Such communication reduces redundancy, supports cross-validation of outputs, and improves the accuracy and robustness of the collective system.

(3) *Coordination and negotiation*: Agents dynamically adjust strategies, resolve conflicts, and balance trade-offs to align with shared objectives. Through these mechanisms, multi-agent systems can adapt to evolving environments and optimize collective decision-making.

Multi-agent collaboration is particularly beneficial for scenarios that demand parallel exploration, multi-perspective reasoning, or the integration of heterogeneous expertise, allowing agentic LLMs to tackle tasks that would be challenging for a single model acting in isolation.

5.4 Routing

Efficiently routing tasks and queries to the appropriate agent or knowledge source is critical for scalability, reliability, and overall system performance. Routing strategies can be understood through several complementary mechanisms.

(1) *Dynamic routing* [35] assigns tasks to agents based on their capabilities, knowledge coverage, or historical performance. By adapting task allocation on the fly, the system ensures that each query is handled by the most suitable agent at a given time.

(2) *Hierarchical routing* [10] employs multi-level controllers to delegate subtasks to specialized agents or modules. This hierarchical structure allows complex tasks to be decomposed and processed efficiently across multiple layers of expertise.

(3) *Load balancing and redundancy* [42] ensures robustness by distributing critical tasks among multiple agents, preventing bottlenecks and providing fault tolerance. Such strategies help maintain consistent performance even under partial system failures.

Formally, routing can be expressed as a function $R : \mathcal{T} \times \mathcal{A} \rightarrow \mathcal{A}'$, where $\mathcal{T}$ denotes the space of tasks, $\mathcal{A}$ represents the set of available agents, and $\mathcal{A}' \subseteq \mathcal{A}$ corresponds to the selected subset of agents responsible for a given task. The goal of routing is to maximize overall system utility—balancing factors such as task success rate, latency, energy consumption, and redundancy—while ensuring coherent knowledge flow among agents.

Beyond these basic mechanisms, routing also interacts closely with memory management and reasoning control. For instance, adaptive routing policies can use feedback from past performance to update task-agent mappings dynamically, thereby enabling meta-learning of optimal routing strategies. Similarly, probabilistic or reinforcement learning–based routers can learn to predict which agent or external resource will yield the most reliable outcome for a given task. By combining internal knowledge, external grounding, multi-agent collaboration, and effective task routing, agentic LLMs move toward truly autonomous and proactive AI systems capable of handling complex, real-world challenges with efficiency and reliability.

6 Applications and Implications

Agentic large language models have the potential to transform a wide range of domains by leveraging their capacity for autonomous reasoning, planning, and interaction. In the context of *scientific discovery*, these models can autonomously explore literature, generate hypotheses, and propose experiments, thereby accelerating research cycles and uncovering patterns that might be overlooked by human researchers. By integrating multi-modal data sources, including text, images, and structured datasets, agentic LLMs support more comprehensive and cross-disciplinary scientific reasoning.

In *decision support*, agentic LLMs facilitate multi-step planning and tool integration to assist with tasks such as travel booking, healthcare recommendations, or business strategy analysis. By combining internal reasoning with external knowledge retrieval, these

systems can provide transparent and explainable recommendations, which is particularly valuable in complex, high-stakes scenarios.

Interactive assistants constitute another important application, where agentic LLMs manage dialogues, query databases, and integrate APIs to deliver context-aware and personalized responses. Such systems can adapt to user preferences over time, enabling tailored experiences in domains ranging from education and legal consultation to customer support.

Finally, agentic LLMs have significant implications for *autonomous systems*. When integrated with sensors and real-time feedback, they can support autonomous decision-making in robotics, logistics, and infrastructure management, offering scalable solutions in dynamic environments.

Despite these promising applications, agentic LLMs raise a number of critical challenges. Ensuring alignment with human values, maintaining safety in high-stakes domains, optimizing computational efficiency, and achieving robustness against adversarial inputs are all essential considerations. These challenges underscore the importance of developing rigorous assurance frameworks to guarantee trustworthy agentic AI. Moreover, ethical considerations, regulatory compliance, and sustainability must be addressed as these systems are deployed at scale. Balancing innovation with accountability will be crucial to maximize the societal benefits of agentic LLMs while mitigating potential risks.

7 Comparative Analysis

The three paradigms of knowledge use in LLMs—internal, external, and agentic—offer complementary strengths and weaknesses. Internal knowledge is advantageous when rapid adaptability is needed, especially in domains where the model's parametric memory suffices. However, it suffers from lack of grounding and verifiability. External knowledge, by contrast, provides transparency and factuality, but its effectiveness is constrained by retrieval quality and latency. Agentic approaches expand the horizon of what LLMs can achieve, enabling complex planning and tool use, yet introduce new challenges of alignment, efficiency, and safety.

Synergies between these paradigms are increasingly important in real-world systems. For instance, agentic LLMs often rely on ICL for reasoning and RAG for grounding, combining the flexibility of internal memory with the factuality of external retrieval. Hybrid systems that dynamically balance between parametric and non-parametric knowledge sources are particularly promising for applications that demand both creativity and reliability.

The trade-offs between these approaches can be characterized along several dimensions. Cost is a major consideration, as retrieval and agentic actions introduce overhead relative to pure ICL. Reliability varies depending on the availability of external resources and the ability to ground outputs in evidence. Scalability is affected by both context length in ICL and retrieval efficiency in RAG. Finally, explainability is often higher in RAG and agentic systems, where outputs can be traced to sources or intermediate steps.

8 Open Challenges and Future Directions

Artificial intelligence [48–50] and machine learning [2, 29] have greatly transformed our society. Despite the remarkable advances in large language models, several open challenges remain in applying these systems to real-world applications. Addressing these challenges is essential for building AI systems that are scalable, reliable, and trustworthy.

(1) *Scaling context versus retrieval* presents a fundamental design consideration. Expanding context windows can enhance in-context learning capabilities by allowing models to directly condition on more examples, yet this approach is memory-intensive and may remain unstructured. In contrast, retrieval-based mechanisms provide a more scalable means of incorporating external knowledge, but they introduce challenges related to retrieval quality, latency, and seamless integration with generative processes.

(2) *Hybrid symbolic-neural reasoning* represents a key frontier for combining the flexibility of deep learning with the precision and interpretability of formal logic. Developing architectures that support structured reasoning, constraint satisfaction, and explainability, while retaining the adaptability of neural methods, remains an open problem that is critical for trustworthy AI.

(3) *Dynamic and persistent memory* is another significant challenge. Current systems primarily rely on ephemeral context windows or static databases, limiting their ability to accumulate experiences or update knowledge over time. Building agents with long-term, verifiable memory raises important questions regarding consistency, version control, and trust, yet it is essential for agents that learn and adapt continuously in real-world environments.

(4) *Evaluation beyond accuracy* is increasingly necessary as models become more autonomous. Existing benchmarks often emphasize surface-level correctness, failing to capture critical dimensions such as factual grounding, reasoning depth, adaptability, efficiency, safety, and trustworthiness. Richer evaluation frameworks and stress tests are needed to assess performance in open-world, high-stakes, and multi-step reasoning scenarios.

Ultimately, progress toward knowledge-centric and trustworthy AI requires unifying these challenges. Future research must not only scale model capacity but also design systems that integrate memory, retrieval, reasoning, and action in ways that are reliable, interpretable, and aligned with human goals, enabling LLMs to operate effectively in complex, dynamic, and real-world environments.

9 Conclusion

This work has reviewed the foundations and frontiers of knowledge use in large language models, focusing on three complementary paradigms: *internal knowledge* through in-context learning, *external knowledge* through retrieval-augmented generation, and *agentic knowledge use* through LLM-based agents. Each paradigm brings distinct strengths—adaptability, factual grounding, and operational autonomy—while also facing unique limitations. By framing these approaches under a unified knowledge-centric perspective, we highlight their synergies as well as their trade-offs. Looking forward, we envision LLMs not merely as passive text generators, but as *knowledge processors* that internalize, ground, and operationalize information. Bridging these paradigms will be key to developing the next generation of trustworthy, explainable, and effective AI systems.

10 Acknowledgment

This material is based upon work supported by NSF awards (SaTC-2241068, IIS-2506643, and POSE-2346158), and a Cisco Research Award.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Yikun Ban, Yunzhe Qi, Tianxin Wei, Lihui Liu, and Jingrui He. Meta clustering of neural bandits. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 95–106, 2024.

[3] Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Chemcrow: Augmenting large-language models with chemistry tools, 2023.

[4] 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. Language models are few-shot learners, 2020.

[5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[6] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2024, page 719–729. ACM, July 2024.

[7] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[8] DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024.

[9] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[10] Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing, 2024.

[11] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning, 2024.

[12] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025.

[13] Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms, 2023.

[14] Aaron Grattafiori .et.al. The llama 3 herd of models, 2024.

[15] Shima Imani, Liang Du, and Harsh Shrivastava. Mathprompter: Mathematical reasoning using large language models, 2023.

[16] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

[17] Chao Jin, Zili Zhang, Xuanlin Jiang, Fangyue Liu, Shufan Liu, Xuanzhe Liu, and Xin Jin. Ragcache: Efficient knowledge caching for retrieval-augmented generation. *ACM Trans. Comput. Syst.*, September 2025. Just Accepted.

[18] Krishnaram Kenthapadi, Mehrnoosh Sameki, and Ankur Taly. Grounding and evaluation for large language models: Practical challenges and lessons learned (survey). In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6523–6533. ACM, August 2024.

[19] Patrick 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. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[20] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Adam Khoja, Zhenqi Zhao, Ariel Herbert-Voss, Cort B. Breuer, Samuel Marks, Oam Patel, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Russell Kaplan, Ian Steneker, David Campbell, Brad Jokubaitis, Alex Levinson, Jean Wang, William Qian, Kallol Krishna Karmakar, Steven Basart, Stephen Fitz, Mindy Levine, Ponnurangam Kumaraguru, Uday Tupakula, Vijay Varadharajan, Ruoyu Wang, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The wmdp benchmark: Measuring and reducing malicious use with unlearning, 2024.

[21] Lihui Liu. HyperKGR: Knowledge graph reasoning in hyperbolic space with graph neural network encoding symbolic path. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China, November 2025. Association for Computational Linguistics.

[22] Lihui Liu. Monte carlo tree search for graph reasoning in large language model agents. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, New York, NY, USA, 2025. Association for Computing Machinery.

[23] Lihui Liu, Yuzhong Chen, Mahashweta Das, Hao Yang, and Hanghang Tong. Knowledge graph question answering with ambiguous query. In *Proceedings of the ACM Web Conference 2023*, 2023.

[24] Lihui Liu, Boxin Du, Heng Ji, ChengXiang Zhai, and Hanghang Tong. Neural-answering logical queries on knowledge graphs. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1087–1097, 2021.

[25] Lihui Liu, Boxin Du, Jiejun Xu, Yinglong Xia, and Hanghang Tong. Joint knowledge graph completion and question answering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 1098–1108, New York, NY, USA, 2022. Association for Computing Machinery.

[26] Lihui Liu, Blaine Hill, Boxin Du, Fei Wang, and Hanghang Tong. Conversational question answering with language models generated reformulations over knowledge graph. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 839–850, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[27] Lihui Liu, Zihao Wang, Ruizhong Qiu, Yikun Ban, Eunice Chan, Yangqiu Song, Jingrui He, and Hanghang Tong. Logic query of thoughts: Guiding large language models to answer complex logic queries with knowledge graphs. *arXiv preprint arXiv:2404.04264*, 2024.

[28] Lihui Liu, Zihao Wang, and Hanghang Tong. Neural-symbolic reasoning over knowledge graphs: A survey from a query perspective. *SIGKDD Explor. Newsl.*, 27(1):124–136, July 2025.

[29] Lihui Liu, Ruining Zhao, Boxin Du, Yi Ren Fung, Heng Ji, Jiejun Xu, and Hanghang Tong. Knowledge graph comparative reasoning for fact checking: Problem definition and algorithms. *IEEE Data Eng. Bull.*, 45(4):19–38, 2022.

[30] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. Tofu: A task of fictitious unlearning for llms, 2024.

[31] Kohei Makino, Makoto Miwa, and Yutaka Sasaki. End-to-end trainable retrieval-augmented generation for relation extraction, 2024.

[32] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023.

[33] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer, 2023.

[34] Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning, 2020.

[35] Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data, 2025.

[36] OpenAI. Gpt-4 technical report, 2024.

[37] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

[38] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

[39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[40] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023.

[41] Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023.
[42] Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets, 2023.
[43] Yixuan Tang and Yi Yang. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries, 2024.
[44] Gemini Team. Gemini: A family of highly capable multimodal models, 2025.
[45] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
[46] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022.
[47] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
[48] Yuchen Yan, Yongyi Hu, Qinghai Zhou, Lihui Liu, Zhichen Zeng, Yuzhong Chen, Menghai Pan, Huiyuan Chen, Mahashweta Das, and Hanghang Tong. Pacer: Network embedding from positional to structural. In *Proceedings of the ACM Web Conference 2024*, pages 2485–2496, 2024.
[49] Yuchen Yan, Baoyu Jing, Lihui Liu, Ruijie Wang, Jinning Li, Tarek Abdelzaher, and Hanghang Tong. Reconciling competing sampling strategies of network embedding. *Advances in Neural Information Processing Systems*, 36:6844–6861, 2023.
[50] Yuchen Yan, Lihui Liu, Yikun Ban, Baoyu Jing, and Hanghang Tong. Dynamic knowledge graph alignment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4564–4572, 2021.
[51] Hui Yang, Sifu Yue, and Yunzhong He. Auto-gpt for online decision making: Benchmarks and additional opinions, 2023.
[52] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.
[53] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023.
[54] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey, 2025.