# Towards Uncovering How Large Language Models Work: An Interpretability Perspective

**Haiyan Zhao**[1] , **Fan Yang**[2] , **Bo Shen**[1] , **Ali Payani**[3] , **Himabindu Lakkaraju**[4] , **Mengnan Du**[1]
[1]New Jersey Institute of Technology [2]Wake Forest University
[3]Cisco Research [4]Harvard University
{hz54, bo.shen, mengnan.du}@njit.edu, yangfan@wfu.edu, apayani@cisco.com, hlakkaraju@hbs.edu

## Abstract

Large language models (LLMs) have shown remarkable performance in tackling natural language tasks, yet the internal mechanisms that enable their impressive generalization and reasoning abilities remain opaque. This lack of transparency presents significant challenges in fundamentally eliminating undesirable behaviors such as hallucinations and toxicity, hindering the safe and beneficial deployment of LLMs. This survey paper aims to uncover the internal working mechanisms underlying LLM functionality through the lens of explainability. First, we review how knowledge is encoded within LLMs via mechanistic interpretability techniques. Then, we summarize what knowledge is embedded in LLM representations by leveraging probing techniques and representation engineering. Additionally, we investigate the training dynamics to explore models' generalization abilities through grokking and memorization. Finally, we explore how the insights gained from these explanations can further enhance LLM performance through model editing, improve efficiency through pruning, and better align with human values.

## 1 Introduction

Large language models (LLMs) have led to tremendous advancements in natural language understanding and generation, achieving state-of-the-art performance in a wide array of real-world tasks. Despite their superior performance across various tasks, the "how" and "why" behind their generalization and reasoning abilities are still not well understood. This lack of understanding poses several challenges. First, LLMs frequently generate hallucinations and factually incorrect output, which complicates efforts to improve their performance. Second, as LLMs become more powerful, problems surrounding potential toxicity, unfairness, and dishonesty threaten to spread misinformation, promote biased or harmful views, and even compromise the safety and wellbeing of society. Therefore, there is an urgent need to fully understand the inner workings of LLMs to address these issues. Gaining insights into how these models operate is a crucial first step towards developing robust safeguards and ensuring their responsible deployment, although our understanding is still in the very early stages.

In this paper, we provide a systematic overview of the existing literature that uncovers the internal working mechanisms of LLMs using explainability techniques (Figure 1). First, we provide a summary of findings on how knowledge is encoded within the architecture of trained LLMs. The explainability technique, mechanistic interpretability (MI), is promising in explaining models' internals at the level of neurons, circuits and attention heads with activations and weights [Saphra and Wiegreffe, 2024]. Second, we examine what knowledge is encoded internally in intermediate representations. To this end, representation engineering (RE) is adopted to explain specific behaviors of models, such as dishonesty, by analyzing hidden/intermediate representations. Specifically, RE focuses on identifying patterns for certain behaviors through probing-based methods and employs them to mitigate undesired behaviors by steering models at inference time [Zou *et al.*, 2023]. Third, we inspect the model training process to understand the development of generalization abilities. Finally, we review how insights from the aforementioned analysis help us improve models in terms of higher performance through model editing, better efficiency through pruning, and better alignment with human values.

Our work differs from existing survey articles on the explainability of LLMs [Zhao *et al.*, 2024; Ferrando *et al.*, 2024], which either summarize explainability techniques or discuss their utilities. In contrast, our goal is to review recent studies that provide insights into trained LLMs and their dynamic training processes. By focusing on model components, representation space, and training processes, we aim to synthesize a line of work particularly focused on uncovering how LLMs function and identify the factors that contribute to their reasoning abilities. Beyond reviewing insights on the inner working mechanisms of LLMs, we further explore how these insights are employed to enhance model performance and benefit humanity.

## 2 Transformer-based LLMs

Before delving into the techniques and insights about how LLMs work, it is essential to establish a foundational understanding of the LLM architecture. This section introduces key components of the decoder-only LLMs, which have been the

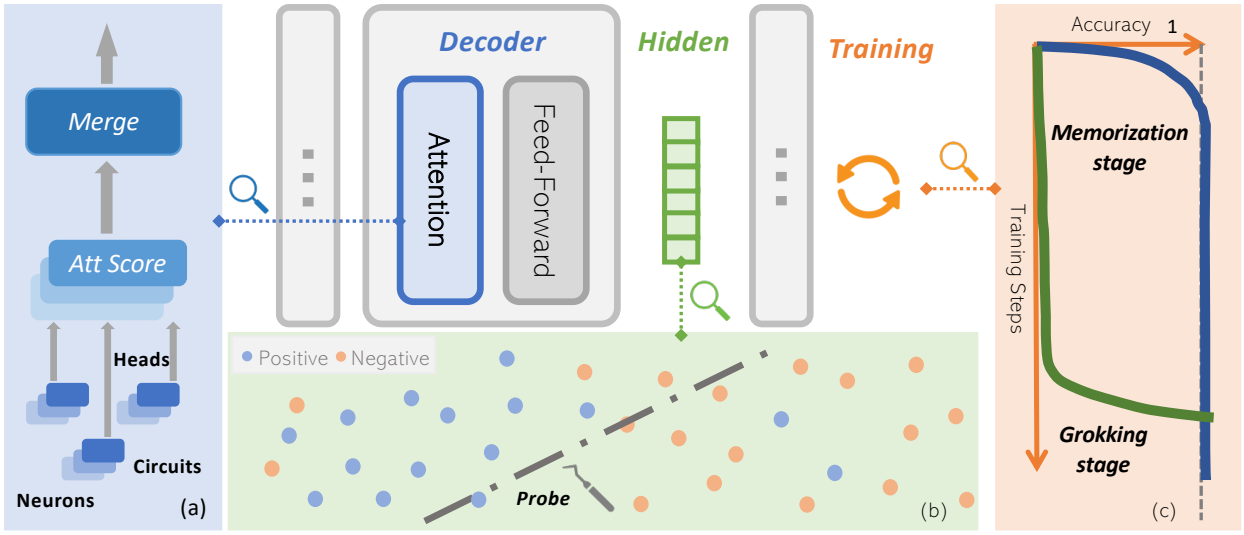



Figure 1: In this work, we review existing progress on how LLMs work, including: (a) how knowledge is encoded within model components; (b) what knowledge is encoded in intermediate representations; and (c) how generalization abilities are achieved during the training process.

dominant architecture for LLMs in recent years [Vaswani *et al.*, 2017].

### 2.1 Decoder-only LLM Architecture

Decoder-only (autoregressive) architecture has been widely used in popular LLMs such as GPT-3.5, GPT-4, Gemma-2, and Llama-3.1. A decoder-only Transformer model typically consists of the following main components (see Figure 2 (a)):

- *Input Embedding Layer*: This layer converts input tokens (words or subwords) into embeddings/representations. Since Transformers don't inherently understand sequence order, positional encodings are added to the input embeddings to provide position information for each token.
- *Multiple Transformer Layers*: The core of the model consists of a stack of identical Transformer layers. Each layer contains: a) Multi-head Self-attention: This allows the model to attend to different parts of the input sequence when processing each token. b) Feed-forward Neural Network: A simple fully connected network. c) Layer Normalization: Applied after each sub-layer to stabilize the learning process. d) Residual Connections: These skip-connections help in training deeper networks by allowing gradients to flow more easily through the model.
- *Output Layer*: The final layer typically projects the representations onto the vocabulary space, producing logits for each token in the vocabulary.

In autoregressive models, input sequences are processed token by token, with each token attending to all previous tokens in the sequence via the self-attention mechanism. During training, the model learns to predict the next token given the previous tokens. At inference time, the model generates each next token by repeatedly sampling from its predicted probability distribution and feeding this predicted token back as input for the next step.

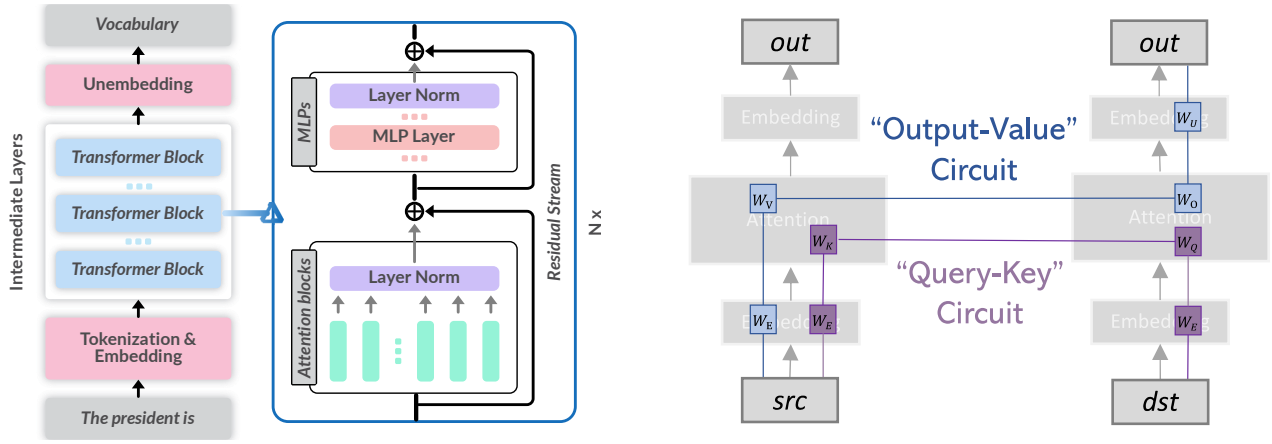
### 2.2 Intermediate Representations

Intermediate representations, also known as hidden representations, are a crucial component in understanding the inner workings of LLMs. The granularity of these representations can vary from the attention head level, where individual heads may specialize in specific patterns, to the layer level, where each layer's output represents a higher-level abstraction of the input. In this work, we refer to intermediate representations as the outputs from "Transformer block" layers (as illustrated in Figure 2 (a)). These layers include both outputs from attention blocks capturing contextual information through self-attention mechanisms, and outputs from feed-forward layers which further transform and process this information.

Understanding these representations is vital as they offer insights into how models transform information at different layers and reveal what types of knowledge or patterns models have learned with probing tools [Belinkov, 2022]. They have also been extensively used in representation engineering to steer model behaviors. In the following sections, particularly in Section 4, we will explore how these representations are employed to interpret world knowledge, factual information, and even unintended biases learned in models.

## 3 How is Knowledge Encoded in Model Architectures?

LLMs' emergent abilities are remarkable, but the underlying mechanisms enabling models to learn vast amounts of knowledge remain unclear. To fully understand LLMs, recent studies have been focused on utilizing MI to reverse engineer LLMs at a more subtle level, such as neurons and attention heads. MI emphasizes understanding the causal mechanisms within models, namely the relationship between model components and their behaviors [Saphra and Wiegreffe, 2024]. In this section, we review studies analyzing functionalities of




(a) Transformer architecture (b) Circuit

Figure 2: (a) The Decoder-only LLM Architecture. (b) An illustration of a Transformer circuit. *src* and *dst* denote previous tokens and the next predicted token respectively. In the toy model, input tokens are first sent to the embedding layer, then pass through the attention block. The intermediate output is further passed into unembedding layer to decode the destination token.

neurons, circuits, and attention heads with toy models and summarize how LLMs process information and make decisions. Notably, foundational discoveries in these simplified models have successfully generalized to explain real-world LLM behaviors, validating concepts like induction heads and driving the development of Sparse Autoencoders (SAEs).

## 3.1 Neurons

Neurons in LLMs are the fundamental units of computation. They receive data from inputs and produce output through activation functions. When a neuron's activation value is high after processing its inputs through the activation function, we consider the neuron to be "activated". Neurons can be activating on multiple unrelated terms (polysemantic) or a single term (monosemantic) [Olah *et al.*, 2020].

### Polysemanticity

Polysemanticity presents a challenge in mechanistically understanding how models operate. *Superposition* appears to be a key cause of polysemantic nature [Olah *et al.*, 2020]. Superposition describes the phenomenon where a feature can be spread across multiple neurons, while a single neuron can simultaneously be involved with multiple features.

Some researchers suggest that superposition originates from an excessive number of features compared to neurons [Olah *et al.*, 2020]. In ReLU networks, Elhage et al. [2022] find that superposition enables the representation of additional features by tolerating some interference. In certain cases, it even facilitates computations such as the absolute value function and logical AND [Hänni *et al.*, 2024]. Others argue that polysemanticity can arise incidentally due to multiple training factors including regularization and neural noise [Lecomte *et al.*, 2024]. Experiments show that feature collisions, introduced through random initialization, consistently result in polysemantic neurons, even when the number of neurons exceeds the number of features [Lecomte *et al.*, 2024]. Another study examines polysemanticity through the lens of the "feature capacity", which denotes the fraction of embedding dimensions consumed by a feature [Scherlis *et al.*, 2022]. Through the analysis of toy models, this work indicates that features are represented based on their importance in reducing loss. More important features are allocated their own dimensions, while less critical ones tend to be polysemantic [Scherlis *et al.*, 2022; Gurnee *et al.*, 2023]. Gurnee et al. [2023] show that early layers tend to represent many features in superposition, while middle layers include dedicated neurons to represent high-level features.

### Monosemanticity

Monosemantic neurons are much easier to interpret. Investigating the factors that enhance monosemanticity is essential for model interpretation. Jermyn et al. [2022] use toy models to reveal that changing the loss minima could improve monosemanticity. Such loss minimum usually coexists with negative biases. However, in practice building a purely monosemantic model is infeasible due to the unmanageable loss. Another line of studies seeks to disentangle superposition to reach a monosemantic understanding using sparse autoencoder (SAE). Instead of focusing on a single neuron, SAE considers layerwise neurons [Cunningham *et al.*, 2024]. It learns sparse activation directions representing monosemantic feature through dictionary learning. However, the effectiveness of this approach remains unclear. Templeton et al. [2024] find that SAEs produce features representing both low-level and high-level concepts, such as code errors and hateful bias-related features. Further, Ferrando et al. [2025] show that SAEs identify features for entity recognition.

## 3.2 Circuits

Circuit analysis was originally proposed to reverse engineer vision models [Olah *et al.*, 2020]. Taking car classification as an example, researchers suggest that some fundamental features such as edge detectors are learned in early layers. These

features are then combined through weights to form circuit units like wheel detectors in later layers. This viewpoint is supported by evidence from a few interpretable circuits performing specific functions such as curve detection and symmetric transformations of basic features, including copying, coloring, and rotation.

However, Transformer models present new challenges with their unique architecture. To address these challenges, a mathematical framework specifically for *transformer circuits* has been proposed [Elhage *et al.*, 2021]. It simplifies the complex architecture of LLMs by focusing on toy models composed of attention blocks with no more than two layers. Typically, two-layer toy models ensure preserving all necessary components of transformer models including input embedding, residual stream, attention layers, and output embeddings, while minimizing architectural complexity.

Attention layers communicate by reading information from the residual stream and then writing their output back. Each attention head operates independently and in parallel, with input embedding matrix $W_E$ and output unembedding matrix $W_U$. These heads consist of key, query, output, and value weights, represented as $W_K$, $W_Q$, $W_O$ and $W_V$. There are two types of circuits: i) "query-key" (QK) circuits formed by $W_Q^T W_K$; ii) "output-value" (OV) circuits composed of $W_U W_{OV}^h W_E$, where $h$ denotes attention head, as shown in Figure 2 (b). The QK circuits are essential for models to recall and retrieve information from earlier context, determining which previous token to copy information from. The computed attention score $W_E^T W_{QK}^h W_E$ indicates how much a destination token attends to a source token. The OV circuits, in turn, determine how the source token influences the output logits [Elhage *et al.*, 2021].

By contextualizing the weights in circuits, research shows that circuits are essentially linear or bilinear functions on tokens. Specifically, Transformers with no layers can model bigram statistics, predicting the next token from the source token. Adding one layer allows the model to capture both bigram and "skip-trigram" patterns. Interestingly, with two layers, Transformer models give rise to "*induction head*" in the second layer and beyond (Section 3.3). These heads are typically composed of heads from their previous layer and are useful in predicting the next token [Elhage *et al.*, 2021].

Beyond the above-mentioned theoretical analysis on toy models, circuits implementing specific functions have been identified in real-world LLMs. For example, a circuit composed of 26 attention heads in GPT-2 small has been found to enable indirect object identification tasks by transmitting information from name tokens to the final outputs. Other research has identified circuits in GPT-2 that perform greater-than computations using a set of MLPs.

### 3.3 Attention heads

A special type of attention head called *induction head* is considered critical in enabling in-context learning abilities within LLMs [Brown *et al.*, 2020], due to their co-occurrence and causal relations [Olsson *et al.*, 2022; Chan *et al.*, 2022]. Induction heads are circuits that complete patterns through prefix matching and copying previously occurred sequences [Olsson *et al.*, 2022]. They comprise two heads: the first attention head from the previous layer attends to previous tokens that are followed by the current token, achieving prefix matching and providing the attend-to token (the token following the current token). The second head copies the attend-to token and increases its output logits. Specifically, this means that if models have encountered patterns such as "[A*][B*]" given the current token "[A]", they can predict "[B]". Beyond the simple example, long prefix matching involving three consecutive tokens has also been observed [Chan *et al.*, 2022]. Consequently, layers with induction heads possess more sophisticated in-context learning abilities than simple copying. However, this theory requires further research on real-world LLMs.

Other functional heads have also been found. For example, Gould et al. [Gould *et al.*, 2024] discovered successor heads that increment tokens like numbers in a natural order. Another study reveals how factual associations are stored and extracted within LLMs [Geva *et al.*, 2023]. A fact association consists of a subject and an attribute. The subject is enriched with subject-related attributes at the early MLP sublayers, which is then propagated to the prediction. The prediction representation "queries" the enriched subject to extract attributes via attention heads. Moreover, Chughtai et al. [2024] suggest that there are also mixed heads containing information from both subject and relation. The subject heads, mixed heads, and relation heads work additively to elicit the outputs.

## 4 What Knowledge is Encoded in Intermediate Representations?

In this section, we present an in-depth review of the knowledge encoded by *LLM representations*, including world knowledge and factual knowledge captured by models. We examine how factors such as layer depth and model scale impact the encoding process.

### 4.1 Probing World and Factual Knowledge

Probing techniques play a crucial role in revealing insights into world knowledge and factual knowledge encoded within models. Specifically, they identify key directions within the representation space that are essential for understanding model behaviors and learned knowledge.

Recent studies have demonstrated that LLMs can learn world models and encode them in their representations for specific tasks. One study successfully uses non-linear probes to uncover Othello board state representations within models [Li *et al.*, 2023]. It demonstrates models' ability to track board states and make predictions without explicit instruction. Furthermore, Nanda et al. [2023b] find that linear representation structures can also perform well on predictions by probing the board state at each timestamp through "my color" and "opponent's color". These findings reveal how models naturally perceive the world, which may differ from human perception. Additionally, by analyzing representations of spatial datasets, Gurnee et al. [2024] demonstrate the model's ability to learn linear representations of space and time across multiple levels. LLMs are also capable of encoding factual knowledge. Marks et al. [2024] craft self-curated true/false datasets to study the geometry of representations of true/false

statements derived from models' residual stream. The data are linearly separable under principal component analysis (PCA). The identified truth directions are used to mediate models' dishonest behaviors locally. Another research direction explores toxicity-related vectors within MLPs through singular value decomposition (SVD). The identified dimensions can be subtracted to achieve efficient mitigation [Lee *et al.*, 2024].

Function vectors have also been discovered within LLMs' attention heads of LLMs, triggering specific task execution across diverse inputs. For example, Todd et al. [2024] find that these function vectors appear in various in-context learning tasks and can execute related tasks even with zero-shot inputs. Additionally, causal interventions at the neuron level help identify individual neurons encoding spatial coordinates and temporal information [Gurnee and Tegmark, 2024].

Lastly, representations has been found to be associated with undesirable LLM behaviors, including dishonesty, toxicity, hallucination, safety concerns. Research has identified specific directions in the representation space that contribute to these behaviors. These directional vectors can be added to the representation space during inference time to guide model behavior without modifying the model's parameters [Zou *et al.*, 2023; Li *et al.*, 2024; Azaria and Mitchell, 2023; Hernandez *et al.*, 2024]. This technique of modifying model behavior by manipulating the representation space is known as representation steering or representation intervention, which provides a lightweight and flexible approach to controlling LLM outputs without the need for model retraining or parameter updates.

### 4.2 Role of Layer Depth and Model Scale

The influence of layer depth on representation analysis has emerged as a compelling research direction. A particularly intriguing finding is the concentration of well-learned knowledge in the middle layers of LLMs. For example, Gurnee et al. [2024] demonstrate that space and time representations achieve optimal quality within the first half of layers across LLMs. Additionally, function vectors with strong causal effects are predominantly found in the middle layers of LLMs, while their effects become negligible in deeper layers [Todd *et al.*, 2024]. Furthermore, research indicates that simpler tasks are mastered in earlier layers, while complex tasks require deeper layers for effective learning [Jin *et al.*, 2025; Ju *et al.*, 2024]. However, the underlying mechanisms of this phenomenon remain unclear.

As predicted by scaling laws, model capabilities increase as models grow larger. Gurnee et al. [2024] show that space and time representations become more precise with model scaling. However, internal mechanisms driving this improved performance during scaling remain poorly understood.

## 5 How is Generalization Ability Achieved During Training?

In this section, we examine how models develop generalization ability during the training process. We focus on two important phenomena associated with generalization: grokking and memorization.

### 5.1 Understanding Grokking

*Grokking* describes a phenomenon where models suddenly demonstrate improved validation accuracy after a period of severe overfitting in over-parameterized neural networks [Power *et al.*, 2022]. This sudden surge in validation accuracy is generally interpreted as the acquisition of generalization ability.

#### A Data Perspective

Experiments on a two-layer decoder-only Transformer network have demonstrated that grokking is influenced by multiple factors, including data characteristics, representations, and regularization. Research shows that smaller datasets require more optimization steps for grokking to occur [Power *et al.*, 2022], while increasing the number of samples reduces the steps needed for generalization [Zhu *et al.*, 2024]. The minimal data requirement for grokking is tied to the threshold number of data points needed to learn robust representations. However, the massive datasets used in LLMs make grokking phenomena less observable [Zhu *et al.*, 2024]. Additionally, regularization techniques can accelerate the onset of grokking, with weight decay proving particularly effective in enhancing generalization capabilities. Wang et al. [2024] have shown that Transformers can develop implicit reasoning abilities exclusively through grokking. Moreover, their research reveals that data distribution plays a more crucial role in achieving grokking than data size alone.

#### Weight Norms

When analyzing models' final layer weight norms during late-stage training, researchers have observed a phenomenon called the *slingshot mechanism*, which occurs only in the absence of regularization. This mechanism is characterized by the simultaneous occurrence of norm growth and training loss spikes, with grokking observed at the onset of these slingshots [Thilak *et al.*, 2022]. The manifestation of both the slingshot effect and grokking can be controlled by modifying optimizer parameters, particularly when using certain adaptive optimizers. However, the universality of this observation across different scenarios remains uncertain.

Another significant finding is the *LU mechanism*, which describes the dynamics between loss and weight norms [Liu *et al.*, 2023]. In algorithmic datasets, researchers have identified an L-shaped training loss curve and a U-shaped test loss reduction relative to weight norms, suggesting an optimal range for weight norm initialization. However, this pattern doesn't readily apply to real-world machine learning tasks, which typically require large initialization values and minimal weight decay. However, Fan et al. [2024] argue that feature ranks and linear probing accuracy may serve as more reliable indicators of phase transition compared to weight norm measurements.

#### Test Loss

*Double descent* describes a pattern in which a model's test accuracy at the log level follows a distinctive trajectory: initial improvement, followed by a decline due to overfitting, and finally a secondary increase as generalization abilities develop [Nakkiran *et al.*, 2020]. This pattern becomes more

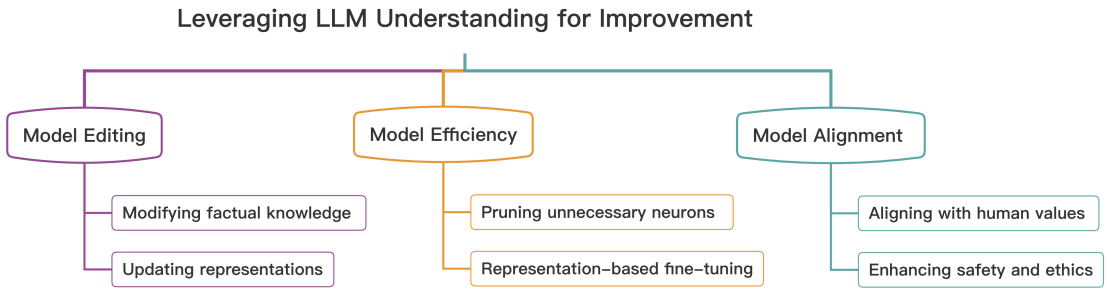



Figure 3: Leveraging understanding of LLMs for targeted improvements. We illustrate three key areas: Model Editing (modifying factual knowledge and updating representations), Model Pruning (improving inference efficiency and reducing model size), and Model Alignment (aligning with human values and enhancing safety and ethics).

pronounced when examining test loss. Researchers have developed a unified framework that integrates grokking and double descent, viewing them as different manifestations of the same underlying process [Davies *et al.*, 2022]. The framework suggests that the transition to generalization stems from slower pattern learning, a perspective supported by subsequent research [Kumar *et al.*, 2024]. This transitional phenomenon has been observed to manifest at both the epoch and model levels.

### 5.2 Memorization

*Memorization* refers to the phenomenon where models rely on statistical features rather than causal relationships for predictions. In LLMs, generalization abilities take precedence over memorization, particularly when evaluating a model's reasoning capabilities. Understanding memorization is therefore crucial for improving model performance.

Studies of two-layer neural models have shown that memorization can exist alongside generalization. Memorization can be reduced through either neuron pruning or regularization techniques. While different regularization methods may have distinct learning objectives, they all contribute to the development of better representations. The terminal phase of training encompasses two distinct stages: i) the grokking process and ii) the decay of memorization learning [Doshi *et al.*, 2024]. However, the underlying mechanisms driving this process remain unclear. Furthermore, the hypothesis that regularization is central to this process has been challenged, particularly given observations of grokking occurring without regularization [Kumar *et al.*, 2024].

Interestingly, recent research hypothesizes that memorization forms an integral phase of grokking [Nanda *et al.*, 2023a]. This study identifies three distinct stages in the grokking process: memorization, circuit formation, and memorization cleanup. Through analysis of model weights, the researchers identified an algorithm using Discrete Fourier Transforms and trigonometric identities to achieve modular addition. Xie et al. [2024] illustrate the coexistence of heavy memorization and improved generalization after fine-tuning. Wang et al. [2025] observe that as model size increases, factual question answering exhibits increased memorization, while machine translation and reasoning tasks show enhanced generalization. Moreover, Menta el al. [2025] argue that memorization primarily occurs in deeper attention heads, while earlier layers play a crucial role in generalizations.

## 6 How to Make Use of The Insights?

Building on the insights from previous sections, we examine how our in-depth understanding of LLMs can be leveraged to enhance their performance through editing, improve efficiency via pruning, and better align them with human values and preferences (Figure 3).

### 6.1 Model Editing for Better Performance

Recent research has made significant strides in model editing. Meng et al. [2022] demonstrate the ability to edit factual knowledge by modifying specific MLP neuron weights, successfully altering neural computations related to factual recall. Stoehr et al. [2024] further reveal that memorized paragraphs can be identified through high-gradient weights in attention heads of lower layers. This research direction focuses on localizing specific attention heads and fine-tuning them to unlearn memorized knowledge, showing promise for enhancing privacy protection in LLMs. However, significant challenges remain. Hu et al. [2024] demonstrate that lifelong knowledge editing often fails due to superposition, potentially affecting unrelated knowledge. Similarly, Gu et al. [2024] identify potential damage to models' general capabilities from editing interventions.

The representation space offers another avenue for model editing, as facts are encoded within these representations. While most current research focuses on modifying representations during inference time, the impact of permanent modifications remains largely unexplored. A breakthrough study proposes a more precise method for editing model representations to alter output distributions [Hernandez *et al.*, 2024]. Rather than merely adding steering vectors during inference, this approach directly modifies entity embeddings to trigger targeted outputs, resulting in shifted entity positions in the embedding space that causally influence model generations.

### 6.2 Improving Model Efficiency

In contrast to deciphering models' inner workings, one study examines the differences between pre-training and fine-tuning phases using MI tools. It reveals that fine-tuning preserves all capabilities learned during pre-training. The trans-

formations between pre-training and fine-tuning arise from "wrappers" in MLPs learned on top of models. These wrappers can be eliminated by pruning a few neurons or retraining on an unrelated downstream task [Jain *et al.*, 2024]. This discovery raises potential safety concerns regarding current alignment approaches.

Unlike pruning neurons, Representation Engineering (RE) directly manipulates representations without requiring optimizations or additional labeled data. RE has also proven effective in model pruning. Several studies show that fine-tuning models with representation engineering can achieve comparable or even better performance than state-of-the-art fine-tuning techniques. For instance, Wu et al. [2024a] employ forward passes from two topics and derive their difference vectors, which are then used during inference without additional fine-tuning. In contrast to conventional parameter-efficient fine-tuning (PEFT), representation-editing-based fine-tuning focuses on learning an additional group of trainable parameters to modify representations directly rather than altering models' parameters. The number of trainable parameters has been reduced by a factor of 32 compared to LoRA [Wu *et al.*, 2024a]. Another approach employs distributed alignment search to find a set of linear subspaces implementing interventions. This method outperforms most PEFT models across various tasks [Wu *et al.*, 2024b].

### 6.3 Model Alignment to Human Values

From a mechanistic perspective, practical applications often evaluate model alignments using various tools. Inspired by induction heads, Yang et al. [2023] measure bias scores of attention heads in pre-trained LLMs by comparing attention scores between biased and regular heads, effectively reducing gender bias through masking identified biased heads. Another study localizes attention heads responsible for deception using linear probing and activation patching. The researchers use intentionally designed prompts to instruct LLMs to be dishonest. Linear probes are trained to classify true/false activations of heads, and selected activations related to dishonest behaviors are then patched with those of honest behaviors to observe output changes. This method successfully locates multiple attention heads across five layers.

Recently, RE has emerged as a promising approach for detecting biases within representation space. A notable study suggests that MLPs operate on token representations to alter the output vocabulary distribution [Geva *et al.*, 2022]. After reverse engineering MLPs, researchers concluded that the output from each feed-forward layer can be viewed as sub-updates to output vocabulary distributions, effectively promoting certain high-level concepts. This insight has been successfully applied to mitigate toxicity levels in LLMs. Another line of research identifies multiple representation vectors within MLPs that encourage undesired model behaviors. These vectors are decomposed using singular value decomposition, enabling researchers to identify specific dimensions that contribute to toxicity.

## 7 Future Research Directions

In this section, we highlight several research directions that warrant further investigation by the research community.

### 7.1 Standardizing Evaluation and Benchmarks

A critical challenge in understanding LLMs is the lack of robust empirical validation for proposed theories like induction heads and circuit analysis, compounded by the absence of standardized evaluation metrics. The field needs agreed-upon criteria for mechanistic explanations and comprehensive interpretability benchmarks [Saphra and Wiegreffe, 2024] to enable systematic comparison of different approaches. However, creating these benchmarks faces unique challenges due to model complexity, absence of ground truth, and diverse architectures. Establishing a rigorous validation framework will require coordinated effort from the research community.

### 7.2 Scaling Interpretability Algorithms

As LLMs grow in size and complexity, scaling interpretability techniques becomes critical. We identify two key challenges: First, we need novel methods to analyze the intricate knowledge structures within LLMs, as current approaches reveal only a fraction of encoded knowledge. Second, existing explainability techniques, including MI tools, are primarily effective on simplified models, creating an urgent need to scale these algorithms to billion-parameter LLMs. This challenge involves both computational efficiency and adapting conceptual frameworks to handle emergent behaviors in large-scale models.

### 7.3 Source of Reasoning Ability

LLMs have demonstrated remarkable reasoning abilities that mirror human cognition, performing complex tasks from multi-step problem-solving to creative thinking. However, we still lack understanding of how these abilities emerge from architectural components and training dynamics. Key research directions to uncover these mechanisms include: 1) examining how LLM components contribute to reasoning processes, 2) analyzing how basic building blocks combine to produce complex reasoning, 3) investigating reasoning development during training, and 4) comparing LLM and human reasoning patterns to identify similarities and differences.

## 8 Conclusions

In this survey paper, we present a comprehensive overview of recent advances in uncovering the inner workings of LLMs through explainability techniques. Our review highlights significant findings about knowledge encoding within LLM architectures, including polysemantic neurons, functional circuits, and attention heads' role in in-context learning. Probing techniques and representation engineering have revealed LLMs' capacity to learn complex world models and encode factual knowledge, while studies of training dynamics have illuminated how generalization abilities emerge. These insights have enabled practical applications in model editing, efficient fine-tuning, and bias mitigation. Despite this progress, challenges persist due to LLMs' complexity and scale, necessitating future work on standardized evaluation metrics, scaling interpretability techniques, and understanding high-level reasoning abilities.

# References


[Azaria and Mitchell, 2023] Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. In *Findings of EMNLP*, pages 967–976, 2023.

[Belinkov, 2022] Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

[Brown *et al.*, 2020] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. Language models are few-shot learners. *NeurIPS*, 2020.

[Chan *et al.*, 2022] Lawrence Chan, Adrià Garriga-Alonso, Nicholas Goldwosky-Dill, Ryan Greenblatt, et al. Causal scrubbing, a method for rigorously testing interpretability hypotheses. *AI Alignment Forum*, 2022.

[Chughtai *et al.*, 2024] Bilal Chughtai, Alan Cooney, and Neel Nanda. Summing up the facts: Additive mechanisms behind factual recall in llms. *arXiv preprint arXiv:2402.07321*, 2024.

[Cunningham *et al.*, 2024] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *ICLR*, 2024.

[Davies *et al.*, 2022] Xander Davies, Lauro Langosco, and David Krueger. Unifying grokking and double descent. In *NeurIPS Workshop ML Safety*, 2022.

[Doshi *et al.*, 2024] Darshil Doshi, Aritra Das, Tianyu He, and Andrey Gromov. To grok or not to grok: Disentangling generalization and memorization on corrupted algorithmic datasets. In *ICLR*, 2024.

[Elhage *et al.*, 2021] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

[Elhage *et al.*, 2022] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, et al. Toy models of superposition, 2022.

[Fan *et al.*, 2024] Simin Fan, Razvan Pascanu, and Martin Jaggi. Deep grokking: Would deep neural networks generalize better? *arXiv preprint arXiv:2405.19454*, 2024.

[Ferrando *et al.*, 2024] Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R Costa-jussà. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*, 2024.

[Ferrando *et al.*, 2025] Javier Ferrando, Oscar Obeso, Senthooran Rajamanoharan, and Neel Nanda. Do i know this entity? knowledge awareness and hallucinations in language models. In *ICLR*, 2025.

[Geva *et al.*, 2022] Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *EMNLP*, 2022.

[Geva *et al.*, 2023] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *EMNLP*, 2023.

[Gould *et al.*, 2024] Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. Successor heads: Recurring, interpretable attention heads in the wild. In *ICLR*, 2024.

[Gu *et al.*, 2024] Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, et al. Model editing harms general abilities of large language models: Regularization to the rescue. In *EMNLP*, 2024.

[Gurnee and Tegmark, 2024] Wes Gurnee and Max Tegmark. Language models represent space and time. In *ICLR*, 2024.

[Gurnee *et al.*, 2023] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, et al. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*, 2023.

[Hernandez *et al.*, 2024] Evan Hernandez, Belinda Z Li, and Jacob Andreas. Inspecting and editing knowledge representations in language models. In *COLM*, 2024.

[Hu *et al.*, 2024] Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Knowledge in superposition: Unveiling the failures of lifelong knowledge editing for large language models. *arXiv preprint arXiv:2408.07413*, 2024.

[Hänni *et al.*, 2024] Kaarel Hänni, Jake Mendel, Dmitry Vaintrob, and Lawrence Chan. Mathematical models of computation in superposition. In *ICML Workshop MI*, 2024.

[Jain *et al.*, 2024] Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P Dick, Hidenori Tanaka, Edward Grefenstette, et al. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. In *ICLR*, 2024.

[Jermyn *et al.*, 2022] Adam S Jermyn, Nicholas Schiefer, and Evan Hubinger. Engineering monosemanticity in toy models. *arXiv preprint arXiv:2211.09169*, 2022.

[Jin *et al.*, 2025] Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, et al. Exploring concept depth: How large language models acquire knowledge at different layers? In *COLING*, 2025.

[Ju *et al.*, 2024] Tianjie Ju, Weiwei Sun, Wei Du, Xinwei Yuan, Zhaochun Ren, and Gongshen Liu. How large language models encode context knowledge? a layer-wise probing study. In *COLING*, 2024.

[Kumar *et al.*, 2024] Tanishq Kumar, Blake Bordelon, Samuel J Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *ICLR*, 2024.

[Lecomte *et al.*, 2024] Victor Lecomte, Kushal Thaman, Rylan Schaeffer, Naomi Bashkansky, Trevor Chow, and Sanmi Koyejo. What causes polysemanticity? an alternative origin story of mixed selectivity from incidental causes. In *ICLR Workshop RA*, 2024.

[Lee *et al.*, 2024] Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin Wattenberg, Jonathan Kummerfeld, and Rada Mihalcea. A mechanistic understanding of alignment algorithms: A case study on dpo and toxicity. In *ICML*, 2024.

[Li *et al.*, 2023] Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *ICLR*, 2023.

[Li *et al.*, 2024] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *NeurIPS*, 36, 2024.

[Liu *et al.*, 2023] Ziming Liu, Eric J Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *ICLR*, 2023.

[Marks and Tegmark, 2024] Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *COLM*, 2024.

[Meng *et al.*, 2022] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *NeurIPS*, 35:17359–17372, 2022.

[Menta *et al.*, 2025] Tarun Ram Menta, Susmit Agrawal, and Chirag Agarwal. Analyzing memorization in large language models through the lens of model attribution. *arXiv preprint arXiv:2501.05078*, 2025.

[Nakkiran *et al.*, 2020] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, et al. Deep double descent: Where bigger models and more data hurt. In *ICLR*, 2020.

[Nanda *et al.*, 2023a] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, et al. Progress measures for grokking via mechanistic interpretability. In *ICLR*, 2023.

[Nanda *et al.*, 2023b] Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In *ACL Workshop BlackboxNLP*, pages 16–30, 2023.

[Olah *et al.*, 2020] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020.

[Olsson *et al.*, 2022] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, et al. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

[Power *et al.*, 2022] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

[Saphra and Wiegreffe, 2024] Naomi Saphra and Sarah Wiegreffe. Mechanistic? *arXiv preprint arXiv:2410.09087*, 2024.

[Scherlis *et al.*, 2022] Adam Scherlis, Kshitij Sachan, Adam S Jermyn, Joe Benton, and Buck Shlegeris. Polysemanticity and capacity in neural networks. *arXiv preprint arXiv:2210.01892*, 2022.

[Stoehr *et al.*, 2024] Niklas Stoehr, Mitchell Gordon, Chiyuan Zhang, and Owen Lewis. Localizing paragraph memorization in language models. *arXiv preprint arXiv:2403.19851*, 2024.

[Templeton *et al.*, 2024] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, et al. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024.

[Thilak *et al.*, 2022] Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon. In *NeurIPS Workshop HITY*, 2022.

[Todd *et al.*, 2024] Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. In *ICLR*, 2024.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[Wang *et al.*, 2024] Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokking of implicit reasoning in transformers: A mechanistic journey to the edge of generalization. In *NeurIPS*, 2024.

[Wang *et al.*, 2025] Xinyi Wang, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, et al. Generalization v.s. memorization: Tracing language models' capabilities back to pretraining data. In *ICLR*, 2025.

[Wu *et al.*, 2024a] Muling Wu, Wenhao Liu, Xiaohua Wang, Tianlong Li, et al. Advancing parameter efficiency in fine-tuning via representation editing. In *ACL*, 2024.

[Wu *et al.*, 2024b] Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. In *NeurIPS*, 2024.

[Xie *et al.*, 2024] Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, et al. Large language interpolators can learn logical reasoning: A study on knights and knaves puzzles. In *NeurIPS Workshop MATH-AI*, 2024.

[Yang *et al.*, 2023] Yi Yang, Hanyu Duan, Ahmed Abbasi, John P Lalor, and Kar Yan Tam. Bias a-head? analyzing bias in transformer-based language model attention heads. *arXiv preprint arXiv:2311.10395*, 2023.

[Zhao *et al.*, 2024] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, et al. Explainability for large language models: A survey. *ACM TIST*, 15(2):1–38, 2024.

[Zhu *et al.*, 2024] Xuekai Zhu, Yao Fu, Bowen Zhou, and Zhouhan Lin. Critical data size of language models from a grokking perspective. *arXiv preprint arXiv:2401.10463*, 2024.

[Zou *et al.*, 2023] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.