big data and cognitive computing

Article

Survey on the Role of Mechanistic Interpretability in Generative AI

Leonardo Ranaldi [1,2]

[1] School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK; l.ranaldi@ed.ac.uk
[2] Human-Centric ART, University of Rome Tor Vergata, Viale del Politecnico, 1, 00133 Rome, Italy

Abstract

The rapid advancement of artificial intelligence (AI) and machine learning has revolutionised how systems process information, make decisions, and adapt to dynamic environments. AI-driven approaches have significantly enhanced efficiency and problem-solving capabilities across various domains, from automated decision-making to knowledge representation and predictive modelling. These developments have led to the emergence of increasingly sophisticated models capable of learning patterns, reasoning over complex data structures, and generalising across tasks. As AI systems become more deeply integrated into networked infrastructures and the Internet of Things (IoT), their ability to process and interpret data in real-time is essential for optimising intelligent communication networks, distributed decision making, and autonomous IoT systems. However, despite these achievements, the internal mechanisms that drive LLMs' reasoning and generalisation capabilities remain largely unexplored. This lack of transparency, compounded by challenges such as hallucinations, adversarial perturbations, and misaligned human expectations, raises concerns about their safe and beneficial deployment. Understanding the underlying principles governing AI models is crucial for their integration into intelligent network systems, automated decision-making processes, and secure digital infrastructures. This paper provides a comprehensive analysis of explainability approaches aimed at uncovering the fundamental mechanisms of LLMs. We investigate the strategic components contributing to their generalisation abilities, focusing on methods to quantify acquired knowledge and assess its representation within model parameters. Specifically, we examine mechanistic interpretability, probing techniques, and representation engineering as tools to decipher how knowledge is structured, encoded, and retrieved in AI systems. Furthermore, by adopting a mechanistic perspective, we analyse emergent phenomena within training dynamics, particularly memorisation and generalisation, which also play a crucial role in broader AI-driven systems, including adaptive network intelligence, edge computing, and real-time decision-making architectures. Understanding these principles is crucial for bridging the gap between black-box AI models and practical, explainable AI applications, thereby ensuring trust, robustness, and efficiency in language-based and general AI systems.

Keywords: artificial intelligence; generative language models; interpretability

check for updates

Academic Editor: Min Chen

Received: 18 February 2025
Revised: 6 June 2025
Accepted: 1 July 2025
Published: 23 July 2025

Citation: Ranaldi, L. Survey on the Role of Mechanistic Interpretability in Generative AI. *Big Data Cogn. Comput.* **2025**, *9*, 193. https://doi.org/10.3390/bdcc9080193

1. Introduction

Artificial intelligence (AI) and machine-learning models have profoundly transformed language understanding, reasoning, and decision making, driving advancements across a wide range of applications. Among these, large-scale neural architectures, particularly transformer-based models, have exhibited exceptional capabilities in handling complex reasoning tasks, generalising across domains, and generating human-like responses. As AI continues to be embedded into networked infrastructures, cloud systems, and the Internet of Things (IoT), its role in optimising distributed processing, intelligent automation, and real-time decision making is becoming increasingly critical. AI-driven models are now at the core of autonomous IoT systems, smart networks, and adaptive communication frameworks, requiring deeper insights into their generalisation mechanisms to ensure robustness and security. Yet, despite their impressive performance on various benchmarks, the fundamental principles, properties, and mechanisms behind their generalisation processes remain poorly understood [1–4]. The black-box nature of these models—further compounded by commercialisation and strategic opacity surrounding their architectures and training methodologies—significantly hinders interpretability [5,6]. This lack of transparency raises concerns about model reliability, accountability, and the potential risks associated with their deployment, particularly in mission-critical AI-integrated networked environments.

A significant challenge stems from their susceptibility to adversarial perturbations and hallucinations, which can lead to outputs that, while plausible, are factually incorrect or internally inconsistent [7]. As AI architectures continue to scale in size and complexity, the need for systematic explainability frameworks becomes increasingly critical, ensuring that models can be effectively analysed, validated, and refined [8–10].

While various explainability-by-design approaches have been introduced to improve model interpretability [11], they often fail to capture the underlying reasoning dynamics or track the evolution of acquired knowledge. Unlike traditional machine-learning models, LLMs transform and encode knowledge in ways that remain fundamentally opaque, requiring deeper investigations into their internal representations, abstraction mechanisms, and knowledge composition [12].

As the need to explore the internal knowledge of LLMs becomes predominant, a crucial technique that has recently emerged is mechanistic interpretability, which allows one to decompose the internal mechanism of the model by identifying and analysing its core components, such as neurons, hidden layers, or attention mechanisms. This analysis enables the investigation of the model's causal capabilities, helping to make them more transparent, interpretable, and reliable. Mechanistic interpretability has the prospect of addressing some of the current issues in the field:

- **Structuring knowledge composition** Unravelling the composition of knowledge in model architectures by investigating the components that enable knowledge acquisition;
- **Acquiring knowledge encoding** Exploring the content of intermediate representations by discerning the acquired knowledge from the behavioural emergent patterns;
- **Emergent abilities** Understanding the mechanisms that lead togeneralisationn in the training process, outlining the boundary withinmemorisationn and the surrounding phenomena.

We investigate the following points by providing a systematic overview of the existing literature (reported in Taxonomy in Figure 1), revealing the mechanisms behind the functioning of LLMs by contributing in the following way:

Firstly, we give some preliminaries of the challenges overcome by the mechanistic interpretability approach (Section 2). Secondly, we proceed to analyse the relevant works that aim to investigate how knowledge is composed in model architectures (Section 3). Hence, by decomposing the functionality of each model component, we aim to interpret how they operate at the level of neurons, circuits, and heads of attention. Thirdly, we examine how knowledge is encoded internally in intermediate representations (Section 4) by providing an in-depth overview of representation engineering's pipelines to explain model-specific behaviour. Finally, having collected a series of evidence supported by the insights of the analyses mentioned above, it can play a strategic role in improving the models in terms of superior performance through internal modifications of the most relevant models (Section 5).

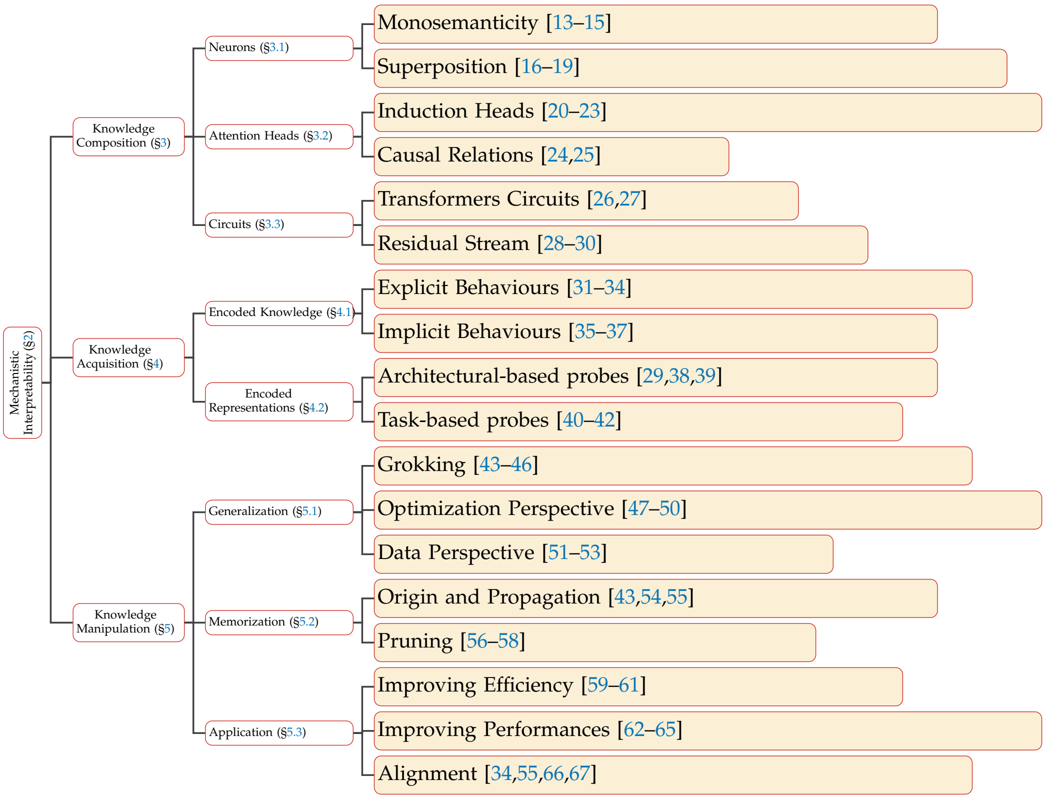
Mechanistic Interpretability (§2)
Knowledge Composition (§3)
Neurons (§3.1)
Monosemanticity [13–15]
Superposition [16–19]
Attention Heads (§3.2)
Induction Heads [20–23]
Causal Relations [24,25]
Circuits (§3.3)
Transformers Circuits [26,27]
Residual Stream [28–30]
Knowledge Acquisition (§4)
Encoded Knowledge (§4.1)
Explicit Behaviours [31–34]
Implicit Behaviours [35–37]
Encoded Representations (§4.2)
Architectural-based probes [29,38,39]
Task-based probes [40–42]
Knowledge Manipulation (§5)
Generalization (§5.1)
Grokking [43–46]
Optimization Perspective [47–50]
Data Perspective [51–53]
Memorization (§5.2)
Origin and Propagation [43,54,55]
Pruning [56–58]
Application (§5.3)
Improving Efficiency [59–61]
Improving Performances [62–65]
Alignment [34,55,66,67]

Figure 1. Taxonomy of approaches that use mechanistic interpretability viewpoint [13–67].

Our contribution significantly extends and complements existing survey articles on the explainability and mechanistic interpretability (illustrated in Table A1), which predominantly focus either on cataloguing explainability techniques [68] or on discussing their importance [69]. Distinct from prior work, we advance the field by critically surveying the emerging body of research dedicated to elucidating the inner workings of LLMs, as exemplified by recent studies [13,43,70,71]. By systematically reviewing and synthesizing research that interrogates the mechanisms underlying LLM behaviour, we go beyond previous accounts to identify and characterize the factors that substantively contribute to LLMs' reasoning abilities. In particular, we highlight how advances in mechanistic interpretability and deliberate control of the training process can drive progress in explainable AI. Moreover, our survey not only consolidates state-of-the-art knowledge on the internal functioning of LLMs but also articulates how these insights may be leveraged to enhance transparency and human-centered understanding. By doing so, we lay the groundwork for the principled enhancement of model interpretability, ultimately fostering the development of safer, more trustworthy, and more effective AI systems.

2. A New Paradigm: Mechanistic Interpretability

As artificial intelligence (AI) continues to evolve, the need for transparency, reliability, and interpretability has become a pressing concern. Modern AI systems, particularly deep-learning architectures, are increasingly deployed in high-stakes environments such as healthcare, finance, autonomous systems, and news recognition, where understanding their decision-making processes is essential [72,73]. Traditional interpretability methods have provided insights into model outputs, but a deeper, more systematic approach is necessary to uncover the inner workings of neural networks and explain how they encode, process, and generalize knowledge.

The mechanistic interpretability paradigm (known as reverse engineering [74]) refers to the activity of investigating artificial neural networks (ANN) to understand the underlying components and mechanisms that determine their behaviour [75]. In fact, examining the inside of neural networks allows one to perceive the rich internal structures of patterns [76]. This approach takes the next step from conventional interpretability methods, as discussed in Section 2. Unconventionally, mechanistic interpretation draws on fields such as neuroscience to study the connections between individual neurons. Thus, considering each neuron as unique and following its weight, an intricate picture emerges of how neural networks function through interconnected circuits that implement meaningful algorithms [26,77]. This mechanistic view enables the deep analysis of artificial neural systems. The individual parts, the neurons, play a comprehensible role, and their circuits of connections implement factual relations about the world. Indeed, it is possible to observe the step-by-step construction of abstract concepts, such as circle detectors, animal faces, cars, and logical operations [75]. A micro-level mechanistic view of LLMs allows for a deeper understanding of their macro-level behaviour. This mechanistic perspective represents a paradigm shift in interpretability, which aims to unpack the causal factors that drive model results.

The Role of Mechanistic Interpretability

The mechanistic view of mechanistic interpretability represents a paradigm shift towards a deeper understanding of the dynamics occurring in ANNs [71,78]. Mechanistic interpretability contributes in the following ways:

- **Implementation Perspective:** Mechanistic interpretability aims to decipher the complexities and mechanisms present in pre-trained models without needing to build models that can be explained explicitly by adapting to existing architectures [79].
- **Vision Perspective:** Mechanistic interpretability brings a global view as it aims to comprehend models as a whole through the lens of high-level concepts and circuits. In contrast, traditional explicability approaches focus on explaining specific predictions made by models, e.g., feature-attribution techniques, which have a limited view of the whole phenomenon [80].
- **Operation Perspective:** Mechanistic interpretability aligns with white-box analysis, as direct access to a model's internal parameters and activations is required. In contrast to black-box explainability tools (e.g., LIME [81] and SHAP [82]), which operate solely based on model inputs and outputs, mechanistic interpretability operates on internal mechanisms by handling unique features.

From what emerges, mechanistic interpretability is a strategic approach that allows for an in-depth understanding of ANNs. It highlights a global and post-hoc standpoint, concentrating on model-specific analysis by interpreting the internal mechanisms and intrinsic logic of complicated systems. This vision drives this approach as necessary for boosting transparency and building trust in LLM, particularly in high-risk scenarios where understanding the underlying motivations of ANN models is as paramount as the decisions themselves.

3. The Knowledge Composition

One of the key points to the success of large language models lies in the complex equilibrium established between extensive training datasets and intricate model architectures [83,84]. Although the practice of releasing open-source model specifications is increasing, the exact mechanisms through which these models acquire and process large amounts of knowledge still need to be discovered. In particular, the exploration of the individual contributions of individual model components and their role in the overall function of LLMs still remains a black-box [85]. Mechanistic interpretability approaches, bypassing the shortcomings of previous methods, as discussed in Section 2, are concerned with proposing methods that enable to interpret LLMs at a granular level, such as neurons (Section 3.1), attention heads (Section 3.2), and circuits (Section 3.3).

3.1. Neurons

The minimal parts within LLMs are the neurons, which are the fundamental building blocks responsible for encoding knowledge and patterns [13]. Similar to non-artificial neural structures, neurons can be activated by several unrelated terms, a phenomenon known as polysemy [75,86]. This feature poses more significant difficulties in mechanistically understanding how patterns work. However, several emerging works [15,18] have introduced key instruments that have shown two structural points in the formation of neural knowledge: Monosemanticity (Section 3.1.1) and Superposition (Section 3.1.2).

3.1.1. Monosematicity

In language artificial learning, it is challenging to distinguish concepts in polysemantic neurons, as various terms can activate them. In contrast, monosemantic neurons, associated with a single concept, are easier to interpret. For this reason, analysing the factors that support monosemanticity is significant for pattern interpretation. Although an emerging line of research proposes several attacks to extract and modify information in a monosemantic manner [14] in real cases, the construction of a purely monosemantic model is not feasible due to the unmanageable loss [15]. To address this problem, another swarm of studies attempts to disentangle the superposition pieces of information (Section 3.1.2) to achieve a monosemantic understanding. The sparse autoencoder architecture is a promising mechanism for this purpose, mainly via dictionary learning where lexicon features are predefined [87]. However, this method has limitations, as its actual functioning is strictly correlated with the network structure and the lexicon's sparsity. Bricken et al. [15] showed that larger autoencoders are able to achieve a finer granularity in feature interpretation and reveal details that cannot be discovered at the low level, i.e., of neurons. These identified features can be used to manipulate the model's output, providing new ways to control and understand the model.

3.1.2. Superposition

Although different features may activate each neuron, one feature may be distributed across several neurons, while another feature may be combined with several features, a phenomenon referred to as overlapping. Elhage et al. [16] argue that this mechanism results from an imbalance between the number of features and the number of neurons [75]. Additionally, the superposition allows for the representation of additional features. To mitigate interference, it is necessary to introduce non-linear filters [16]. However, with sparse input features, the superposition effectively supports the representation of these features and allows for calculations such as the absolute value function [16]. Neurons within models can be monosemantic or polysemantic.

Scherlis et al. [17] investigates polysemanticity through the lens of feature capacity, indicating the fraction of embedding dimensions consumed by a feature in the representation space. This work suggests that features are represented following their importance in loss reduction. The most essential features are assigned their dimensions, while the less critical ones may be neglected, and the others share the embedding dimensions. Features end up sharing dimensions only when the assignment of additional capacity does not result in a loss of [17]. Furthermore, the relationship between overlap and feature importance has been demonstrated on LLM [18]. The experiments show that the first layers tend to represent many overlapping features, whereas the middle layers include neurons dedicated to representing high-level features.

On the other hand, Lecomte et al. [19] have shown that polysemanticity occurs incidentally due to factors encountered during the training process, such as regularization and neural noise. In particular, downstream of formal demonstrations in [19], it was shown that a constant fraction of feature collisions, introduced through random initialization, can always result in polysemantic neurons, even when the number of neurons exceeds the number of features.

3.2. Attention Heads

The in-context learning abilities of LLMs are closely associated with a strategic mechanism within the transformer architecture [20,21]. This mechanism, referred to as the induction head, plays a pivotal role in enabling pattern completion through prefix matching and the replication of previously encountered sequences [21]. Induction heads comprise two principal components: the first, inherited from the preceding layer, attends to prior tokens up to and including the current token, thereby achieving prefix matching and identifying the ′attend-to′ token (i.e., the token that immediately follows the current token). The second component—the induction head proper—copies this attend-to token and amplifies its corresponding output logits.

In practice, this mechanism enables models that have previously observed patterns, such as [A*][B*], to predict [B] when presented with the current token [A] [21]. While early demonstrations of this phenomenon focused on single-token correspondences in controlled settings, subsequent research has shown that induction heads can generalize to longer prefixes, successfully matching and replicating sequences spanning multiple consecutive tokens [24].

Hence, layers with induction-heads reveal more emergent in-context learning abilities than simple copying activity [88]. Moreover, several parallel works have demonstrated the causal relationships between induction heads and in-context learning abilities by observing the change in in-context learning abilities after manipulating induction heads [21,23]. Although this theory comprehensively explains the mechanisms behind the transformer with only a few attention layers, further ablation studies are still needed to validate its effectiveness. For this reason, it is essential to note that this framework is exclusively based on attention heads and does not incorporate MLP layers [22].

3.3. Circuits

Circuits represent a foundational concept within the field of mechanistic interpretability. Their origins lie in the reverse engineering of vision models, where individual neurons and their interconnections are examined as functional units [75]. Numerous studies have demonstrated that features at the lower levels of such models act as internal units—for instance, edge detectors—which are subsequently integrated through weighted connections to form higher-order circuit components. This framework is particularly evident in the identification of circuits comprising interpretable neurons that execute well-defined functions, such as curve detection [89] or frequency discrimination [90]. Moreover, a variety of symmetrical transformations of basic features—including copying, scaling, flipping, coloring, and rotation—can be instantiated by fundamental neurons, often referred to as equivariant units or motifs [91].

Although there is extensive literature in computer vision supported by rich insights into vision models, transformers present new challenges with their unique architecture, distinguished by attention blocks. A specific mathematical framework for transformer circuits has been proposed [27]. This framework facilitates the complex architecture of LLM circuits, focusing on decoder-only transformer models with no more than two layers, all of which are composed entirely of attention blocks. In this toy model, the transformer incorporates input embedding, residual flow, attention layers, and output embeddings. The attention layers read information from the residual stream and then write their outputs into it. Therefore, communication takes place through read-and-write operations at the layer level [30].

Each attention head works independently and in parallel, contributing its output to the residual stream [28]. These components consist of key, query, output, and value vectors, represented as W_K, W_Q, W_O, and W_V. There are two types of circuits: (i) "query-key" (QK) circuits; (ii) "output-value" (OV) circuits [27]. The QK circuits, formed by $W_Q^T\, W_K$, are essential in resolving which previously learned token to duplicate information from [27]. It is indispensable for models to recall and retrieve information from previous contexts. In contrast, the OV circuits, composed of $W_O\, W_V$, decide how the current token influences the output logits.

The downstream result shows that transformers with no layer can model bigram statistics, predicting the next token from the source token. Adding one layer allows the model to capture n-gram patterns. Interestingly, with two layers, transformer models give rise to a concept termed as induction-head. These induction heads exist in the second layer and beyond. Usually, they are comprised of heads from their previous layer, which allows suggesting the next token based on the current ones [27].

4. Acquiring Knowledge

Mechanistic interpretability centers on the investigation of structural model components to formalize and manipulate the processes by which models acquire knowledge. Building on the architectural foundations outlined in Section 3, this section presents an overview of existing studies that have examined the architectural elements of artificial neural networks, with particular emphasis on transformer-based LLMs. Here, we provide an in-depth analysis of the knowledge encoded within LLM representations, encompassing both world knowledge and factual information internalized by these models. Special attention is given to the influence of layer depth and model scale on the nature and extent of this knowledge encoding.

4.1. The Seek of Knowledge

Different probing techniques investigate the dynamic structure of representations constructed by LLMs to understand whether they encode factual and world knowledge. In particular, probing techniques identify specific directions within the representation space that are essential for understanding certain behaviours and encoding knowledge [31–33]. Recent studies have claimed that LLMs can learn factual representations of the world and consequently encode them into representations for specific tasks. A series of works [35,38] have displayed models' ability to track the board state and construct predictions without being explicitly told. Li et al. [38] uses non-linear probes to reveal world representations within models, specifically in the context of the game of Othello that Hazineh et al. [35] used to reveal that the analysed model contains representations that can orient decision and habitual processes causally. Later, Nanda et al. [92] found that linear representation structures can also perform well in forecasting by simply changing the expression of the card state at each timestamp. The linear and non-linear explanations reveal how models naturally perceive the world, which may differ from humans. Moreover, by investigating representations of spatial datasets, Gurnee and Tegmark [39] exposes the model's proficiency in learning linear representations of space and time across multiple levels.

Concurrently, these models are capable of encoding factual knowledge that is more tangible and concrete than abstract forms. Marks and Tegmark [36] construct self-corrected binary datasets to investigate the geometry of representations of true and false information derived from a model's residual stream. By applying principal component analysis, a clear linear structure is revealed, wherein specific directions can be leveraged to mitigate the model's unfair behaviour locally. In contrast, Burns et al. [37] demonstrates that it is possible to distinguish knowledge from model behaviour by analysing the internal representations acquired through learning.

The manipulation of internal states has also been extensively investigated in Lee et al. [34], Li et al. [93], which have explored toxicity-related vectors within MLP blocks through singular value decomposition, thus proposing effective mitigation techniques and detoxification strategies.

Following the line of research on detoxification and the manifestation of hallucinations, another branch of work aims to extract these undesirable behavioural patterns. In particular, leveraging the direction in the representation space is identified as contributing to a specific behaviour. This directive will then modify the representations that models' behaviours can be controlled [32]. For example, Li et al. [94] employs this technique to probe and improve the honesty of models. Azaria and Mitchell [95] also successfully distinguishes the truthfulness of statements by simply introducing a classifier on model representations. Recent work has been developed to identify hallucination tokens in responses by integrating a range of classifiers trained on each layer from separate hidden parts, including MLPs and attention layers [96].

Function vectors have also been discovered within the attention heads of LLMs, which activate the execution of a specific task across diverse types of inputs. Todd et al. [29] discovered that these function vectors are shown in different in-context learning tasks and can execute corresponding tasks despite zero-shot inputs. Furthermore, causal interventions at the neuron level can aid in determining the particular neurons encoding spatial coordinates and time facts [39].

4.2. Role of Layer Depth and Model Scale

The influence of layer depth and model scale on internal representations has emerged as a central focus in recent interpretability research. Empirical findings indicate that a broad spectrum of knowledge tends to be robustly encoded within the intermediate layers of large language models. Gurnee and Tegmark [39] shows that spatial and temporal representations reach the best quality up to the middle of the layers in open-source LLMs. Similarly, the function vectors with substantial causal effects are also gathered from the middle layers of LLMs, while the effects are near zero in the more profound layers [29]. Evidence from recent studies indicates that distinct layers specialize in learning different types of concepts: lower layers tend to capture simpler, surface-level patterns, while deeper layers are required to encode more abstract and complex tasks [40,42]. However, the underlying reason why the middle layers perform so well remains to be explored. It is stated that more outstanding capabilities are generally acquired as models increase in scale, as discussed in [97]. Furthermore, spatial and temporal representations are more accurate as the scale of the models increases [39]. However, when the scale of the models remains unspecified, the internal mechanism leads to better performance.

5. The Boundary Position: Generalization vs. Memorization

The mechanistic interpretability paradigm (Section 2) is defined by a number of activities that play a key role in understanding the inner dynamics of artificial neural networks

(ANN), in this particular case transformer-based large language models (LLMs), from both an architectural (Section 3) and an internal encoding acquisition (Section 4) perspective.

In practice, the functionalities outlined above extend to the analysis of model training procedures. We therefore review state-of-the-art studies that employ mechanistic interpretability techniques to delineate the boundary between generalization and memorization—two capabilities that often intertwine during training. In particular, we focus on two central phenomena: memorization and a paradigmatic instance of generalization known as grokking. The latter refers to the point at which models, having initially overfitted, suddenly exhibit a marked improvement in validation accuracy, highlighting the complex interplay between memorization and generalization. analysing grokking offers valuable insights into the emergence of generalization during training. Additionally, a detailed examination of memorization, where models rely predominantly on statistical patterns rather than causal relationships, can further clarify the respective roles of generalization and memorization in shaping model behaviour.

5.1. Generalization Beyond Memorization

The phenomenon of grokking lies in the fact that models suddenly improve validation accuracy behind extreme overfitting on hyperparameterized artificial neural networks [43–45]. Accordingly, the gain in validation accuracy is interpreted as an increase in generalization capability. The empirical reasons for this phenomenon are studied from the perspective of optimization and evaluation algorithms (Section 5.1.1) and from a data perspective (Section 5.1.2).

5.1.1. Optimization Perspective

The slingshot effect indicates the occurrence of grokking [48,98–100]. During the study of weight norms of the final layers in models that do not use regularization techniques, this effect seems to emerge [46,101]. In particular, the slingshot effect describes cyclic behaviour during the terminal phase of training, where oscillations between stable and unstable rules occur (training loss spike). The spike co-occurs with a phase where weight norms grow, observed by a phase of norm plateau. Thilak et al. [98] point out that grokking, non-trivial component adaptation, appears only at the beginning of the slingshots effect. The slingshot effect and grokking formation can be modulated by altering the optimizer parameters precisely when operating adaptive optimizers such as Adam [102]. However, whether this observation holds universally across various scenarios is yet to be determined.

Similarly, Liu et al. [47] started to analyse the loss landscapes of ANN. The mismatch between training and test loss landscapes is the cause of grokking, defining it as the LU mechanism. In algorithmic datasets, an L-shaped training loss and a U-shaped test loss reduction concerning weight norms are identified, suggesting an optimal coverage for initializing weight norms [47,49]. Moreover, this finding only seamlessly transfers to real-world machine-learning tasks, where extensive initialization and minor weight decay are continually needed. Earlier works attribute it to a match between the early-phase implicit bias preferring kernel predictors generated by large initialization and a late-phase inferential bias leaning min-norm/margin predictors promoted by minor weight decay [103,104]. Correspondingly, Merrill et al. [105] supposes this match displays a challenge between a dense subnetwork in the initial step and a sparse one behind grokking.

However, from a more model-loss-centered view, as stated in [106], double descent captures the pattern in which the test accuracy of a model at the log level initially improves, then decreases due to overfitting and finally increases again after gaining the ability to generalize. This effect is evident in the test loss. A unified framework was designed to combine grokking and double descent, treating them as two representations of the same underlying process [107]. The framework attributes the change in generalization to slow pattern learning, further supported by Kumar et al. [57]. Later contributions demonstrate that this transition is displayed both at the level of epochs and patterns [50].

5.1.2. Data Perspective

On the other side of the coin, while algorithmic factors are undoubtedly important, a substantial body of research has examined the pivotal role of data in the learning process. Notably, studies conducted on two-layer, decoder-only transformer models indicate that grokking is intricately linked to data availability, the nature of learned representations, and regularization strategies. For instance, smaller datasets necessitate a greater number of optimization steps before grokking can occur [44], whereas increasing the number of training samples reduces the steps required for generalization to emerge [52]. Liu et al. [51] contend that the minimum data required for grokking corresponds to the smallest subset sufficient to establish robust representations, further demonstrating that the onset of generalization frequently coincides with the formation of well-structured embeddings. Moreover, regularization interventions—particularly weight decay—have been shown to expedite grokking and reinforce generalization capabilities. However, recent studies challenge these earlier findings. In particular, Zhu et al. [52] and Chen et al. [108] present evidence that, in the context of large-scale datasets, grokking becomes only marginally feasible in LLMs. Additionally, while transformers are capable of acquiring implicit reasoning abilities, this typically requires protracted training well beyond the point of overfitting [53].

5.2. Memorization

Although generalization is supported by emergent phenomena such as grokking (Section 5.1), parallel, often introverted series of phenomena occur that models predict with statistical features rather than causal relationships phenomena best exemplified as memorization. The study using slightly corrupted algorithmic datasets with two-layer neural models has demonstrated that memorization can coexist with generalization. Furthermore, memorization can be mitigated by pruning relevant neurons or by regularization [56]. Although different regularization methods might not share learning objectives, they all contribute to more reasonable representations. The training process in the analysis consists of two stages: first, there is the grokking process and then the decay of memorization learning [56]. However, the underlying causalities behind this process have not been fully comprehended. Similarly, the hypothesis that regularization is the key to this process is under discussion, mainly in light of observing grokking in the absence of regularization [57]. The significance of the rate of feature learning and the number of necessary features is favored in explanations, questioning the role of the weight norm [57].

Nanda et al. [43] hypothesizes that memorization comprises a step of grokking. The analysis finds that grokking includes three stages: memorization, circuit formation, and memorization cleanup. Moreover, Nanda et al. [43] specifies an algorithm that utilizes discrete Fourier transforms and trigonometric identities to achieve modular addition by analysing the model's weights. The circuits enabling this algorithm evolve steadily, rather than randomly walking. However, our understanding of the relationship between memorization and grokking still needs to be improved.

5.3. Application

The mechanistic interpretability paradigm (Section 2) delivers tools for exploring architectural knowledge (Section 3), coding, and learning knowledge representations (Section 4). This understanding enables the analysis of phenomena that emerge during and behind training (Section 5). Gathering these insights can be harnessed to improve the deep understanding of LLMs mechanisms, improving their efficiency (Section 5.4), empowering their performance (Section 5.5), and better aligning them with human values and preferences (Section 5.6) by reducing the ongoing gap between humans and models.

5.4. Improving Efficiency

Attention heads and neuron activations play a fundamental role in the architecture of transformers [25]. Hence, causal tracing and analysis of causal mediation from mechanistic interpretability is a fundamental technique for deep understanding of models. Stolfo et al. [59], Hou et al. [60] study the importance of the attention mechanism in revealing how the model processes input, showing that this mechanism enables models to extract query information in the final token in the first levels of transformer-based models. In addition, the result information is incorporated into the residual stream in the last MLP levels. Localizing this information improves fine-tuning on specific tasks [61] and allows one to focus on only a part of the models while ignoring information that is superfluous to the specific task (pruning technique) [109–111].

However, these studies are merely specializations of the study proposed in [112] that examines the differences between the pre-training and fine-tuning phases with mechanistic interpretability tools. Jain et al. [112] show that fine-tuning retains all the skills learned in the pre-training phase. Transformations between pre-training and fine-tuning result from wrappers located in the MLPs learned at the top of the models. In particular, wrappers can be eliminated by pruning some neurons or retraining an unrelated downstream task. This finding sheds light on potential security problems associated with current alignment approaches.

By exploiting these attacks, representation engineering aims to directly manipulate representations without needing optimization or further labeled data [32]. This technique has proven effective, specifically as a specialization of model pruning. Wu et al. [110] propose techniques for fine-tuning models with representation engineering and performing similar and even more reasonable performance than more evolved fine-tuning methods, as in [41,113]. After that, Wu et al. [110] reveals the feasibility of fine-tuning models via editing representations. Unlike conventional parameter-efficient fine-tuning, representation editing focuses on learning more trainable parameters to change representations directly other than models' parameters. The trainable parameters have been reduced to a factor of 32, reaching that of LoRA [109]. Geiger et al. [114] employs distributed alignment tracking to find a set of linear subspaces implementing interventions. This strategy outperforms most PEFT models on different scenarios [41,58].

5.5. Improving Performances

As well as efficiency, tangible results were also observed in terms of performance. Cao et al. [115] demonstrated that it is feasible to edit factual knowledge by changing the weights of specific neurons in MLPs. Meng et al. [31] adopts this approach by changing neural computations connected to recall factual knowledge. Afterward, they expand this method to allow multiple edits simultaneously [31]. Although these methods are effective for targeted edits, their ability to edit relevant knowledge and control forgetting still requires further research [62].

Stoehr et al. [64] suggest that the sections memorized by a model can be pinpointed using high-gradient weights in the attention heads of the lower layers. This research employs localization techniques to identify detailed attention heads, which are fine-tuned to unlearn the memorized knowledge. This approach aims to improve privacy protection in LLMs, although an exhaustive evaluation is yet to be conducted.

In particular, facts are encoded in the representation space, making assembling representations a natural contender for editing models' outputs. So far, most analyses focus on modifying representations at inference time, while the influence of permanent modifications has yet to be studied. Recent work provides a more precise way to edit model representations to change their output distributions [63,116]. Rather than only counting the derived vectors into effects representations, this investigation directly adjusts the embedding of a related entity to trigger targeted outputs. Therefore, the modified entity's position in the embedding space has changed, leading to a causal effect on model generations.

5.6. Mechanistic Interpretability to Refine Models' Capabilities

From a mechanistic perspective, various practical applications have been proposed to enhance and refine human model alignment. Particularly in the case of bias, where current measures are based on probing, prompts designed to elicit specific responses are primarily exploited, known as prompt engineering. The completeness of these prompts determines the effectiveness of these measures. However, prompts can only capture recognized biases using a finite set of examples, often confined to specific tasks. In this way, biases that have been learned but remain unacknowledged and unaddressed across generations cannot be detected.

To address these issues, several novel works have been approaching the problem from a mechanistic point of view [54,55,65,117]. Zhang and Nanda [117] aligned the guidelines previously used Campbell et al. [118] to locate the attention heads responsible for lying with a linear survey and an activation patch. Yang et al. [66] focused on stereotype recognition estimates bias scores of attention heads in pre-trained LLMs. They implemented a method to ensure the accuracy of determining biased heads by comparing the changes in attention scores between biased and regular heads.

Representation engineering has emerged as a promising avenue for detecting biases within the embedding space. Sharma et al. [65] suggest that MLPs operate on token representations to alter the distribution of output vocabulary. After reverse engineering, the output from each feed-forward layer can be seen as sub-updates to output vocabulary distributions, essentially promoting certain high-level concepts that could effectively mitigate toxicity levels in LLMs. Lee et al. [34] identified multiple representation vectors within MLPs that encourage models' undesired behaviours. Hence, they decomposed the vectors using singular value decomposition, enabling them to pinpoint specific dimensions that contributed to toxicity. Finally, Jin et al. [67] interpreted the mechanism of knowledge conflicts through the lens of information flow and mitigated conflicts through precise, systematic interventions.

6. Final Discussion and Future Challenges

This paper investigates two emerging paradigms at the forefront of explainability—mechanistic interpretability and representation engineering. We present a comprehensive overview of how knowledge is architecturally composed and internally represented within large language models (LLMs). Our analysis examines the underlying dynamics that delineate the boundary between memorization and generalization. We highlight how insights from these approaches can be leveraged to enhance LLM performance through targeted model editing, increase efficiency via pruning strategies, and promote better alignment with human values and preferences. There is some preliminary progress in understanding the inner workings of LLMs, but upon closer examination, multiple challenges have emerged. Although LLMs have encoded a significant amount of real-world knowledge, current research has revealed only a small portion of the encoded knowledge. Hence, future efforts should develop scalable techniques to effectively analyse and interpret the intricate knowledge structures embedded in models. LLMs demonstrated remarkable reasoning capabilities by displaying human-like cognitive capacities. However, the current understanding of how high-level reasoning capabilities emerge from the interaction of architectural components and training dynamics requires improvement. Further research is required to uncover the complex mechanisms that give rise to advanced reasoning capabilities in LLMs. In sum, insights from mechanistic interpretability and representation engineering have established a foundational basis for progress in model editing, pruning, and alignment. However, the progress made thus far has been relatively modest. It is therefore of utmost importance that we continue to build on these insights and develop techniques that can significantly enhance the performance of LLMs.

7. Conclusions

A thorough understanding of the internal mechanisms that underpin generalization and reasoning in large language models (LLMs) remains an open challenge. This paper surveys the current landscape of explainability approaches dedicated to investigating the emergent mechanisms within LLMs. Adopting a mechanistic interpretability perspective, we examine the strategic architectural components believed to underpin these advanced capabilities. We move to evaluate methodologies for quantifying the knowledge acquired and expressed by LLMs, with distinct attention to the composition and encoding of knowledge within model parameters. This analysis draws on key developments in both mechanistic interpretability and representation engineering. Subsequently, we review applications from a mechanistic standpoint, focusing on the explanation of emergent phenomena in training dynamics—such as grokking—which illuminate the processes underlying generalization in LLMs. Finally, we consider how insights from mechanistic analysis can be leveraged to enhance LLM performance through model modification, efficiency improvements, and better alignment with human preferences. While preliminary advances in areas such as model editing, pruning, and alignment have been facilitated by mechanistic interpretability, further research is required to fully capitalize on these insights and realize their potential for advancing LLM performance.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Comparative summary of major papers on the intersection of explainability and mechanistic interpretability in NLP and LLMs, highlighting covered methods, strengths, weaknesses, and principal applications.

Paper	Methods Covered/Categorization	Strengths	Weaknesses	Main Applications/Use Cases
[65]	Local/Global, Self-explaining/Post-hoc; Reviews feature importance, surrogate models, example-driven, provenance-based, declarative induction; Summarizes explainability and visualization techniques	Connects operations, visualization, and evaluation; Lists representative papers for each technique; Points out gaps and challenges	Focus is broad, not LLM-specific; Limited depth on transformer-based models;	Explanation of NLP model predictions; Resource for model developers; Trust-building in NLP systems

Table A1. *Cont.*

Paper	Methods Covered/Categorization	Strengths	Weaknesses	Main Applications/Use Cases
[69]	Organises XAI challenges and future directions; Thematic: General, Design, Development, Deployment; analyses multidisciplinary aspects and societal, regulatory, user needs	Systematic, multi-level challenge analysis; Addresses regulatory and human-centric perspectives;	Not specific to NLP or LLMs; Less focus on concrete methods; Little technical comparison of approaches	Identifying open research challenges; Guiding research agendas; Human-AI trust, regulatory compliance
[70]	Classifies LLM explainability: Local vs Global; Local: Feature attribution, perturbation, gradient-based, vector-based; Global: Probing, mechanistic interpretability, circuit discovery;	Focused on LLMs/transformers; Critical evaluation of strengths/limits for each method; Discusses evaluation metrics and datasets; Connects explainability to model enhancement (editing, alignment)	Some coverage is high-level; Ongoing field: not all methods are mature; Few real-world benchmarks for all techniques	LLM transparency; Model editing and control; Reducing hallucination, improving alignment
[12]	Reviews interpretability in neural NLP; Structural analysis (probing classifiers);	Explains main lines of analysis with examples; Discusses limitations and future directions; Covers both theory and hands-on tools	Lacks systematic empirical comparison; More pedagogical than evaluative; Pre-dates some LLM developments	Education for NLP/ML community; Foundation for newcomers; Understanding neural network behaviour in NLP
[78]	Structured taxonomy of mechanistic interpretability methods; Covers observational, interventional, intrinsic and developmental interpretability; Strong focus on MI for AI safety and alignment	Methods and safety motivations; Discusses challenges and future directions; Connects MI to societal/AI safety needs	Primarily focused on technical MI, less coverage of global/external XAI methods; Societal and regulatory implications are more briefly addressed; Open challenges remain in scalability and full automation	Mechanistic analysis of neural networks; Model transparency and AI safety; Reverse engineering model behaviour;

References

1. Marques, N.; Silva, R.R.; Bernardino, J. Using ChatGPT in Software Requirements Engineering: A Comprehensive Review. *Future Internet* **2024**, *16*, 180. [CrossRef]
2. Ranaldi, L.; Pucci, G. Knowing Knowledge: Epistemological Study of Knowledge in Transformers. *Appl. Sci.* **2023**, *13*, 677. [CrossRef]
3. Peng, J.; Zhong, K. Accelerating and Compressing Transformer-Based PLMs for Enhanced Comprehension of Computer Terminology. *Future Internet* **2024**, *16*, 385. [CrossRef]
4. Ranaldi, L.; Fallucchi, F.; Zanzotto, F.M. Dis-Cover AI Minds to Preserve Human Knowledge. *Future Internet* **2022**, *14*, 10. [CrossRef]
5. Gifu, D.; Silviu-Vasile, C. Artificial Intelligence vs. Human: Decoding Text Authenticity with Transformers. *Future Internet* **2025**, *17*, 38. [CrossRef]
6. Li, J.; Maiti, A. Applying Large Language Model Analysis and Backend Web Services in Regulatory Technologies for Continuous Compliance Checks. *Future Internet* **2025**, *17*, 100. [CrossRef]
7. Petrillo, L.; Martinelli, F.; Santone, A.; Mercaldo, F. Explainable Security Requirements Classification Through Transformer Models. *Future Internet* **2025**, *17*, 15. [CrossRef]
8. Tenney, I.; Das, D.; Pavlick, E. BERT Rediscovers the Classical NLP Pipeline. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Korhonen, A., Traum, D., Màrquez, L., Eds.; pp. 4593–4601. [CrossRef]
9. Lampinen, A.; Dasgupta, I.; Chan, S.; Mathewson, K.; Tessler, M.; Creswell, A.; McClelland, J.; Wang, J.; Hill, F. Can language models learn from explanations in context? In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; Goldberg, Y., Kozareva, Z., Zhang, Y., Eds.; pp. 537–563. [CrossRef]
10. Aggrawal, S.; Magana, A.J. Teamwork Conflict Management Training and Conflict Resolution Practice via Large Language Models. *Future Internet* **2024**, *16*, 177. [CrossRef]
11. Babaey, V.; Ravindran, A. GenSQLi: A Generative Artificial Intelligence Framework for Automatically Securing Web Application Firewalls Against Structured Query Language Injection Attacks. *Future Internet* **2025**, *17*, 8. [CrossRef]

12. Belinkov, Y.; Gehrmann, S.; Pavlick, E. Interpretability and Analysis in Neural NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*; Savary, A., Zhang, Y., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2020; pp. 1–5. [CrossRef]
13. Golechha, S.; Dao, J. Position Paper: Toward New Frameworks for Studying Model Representations. *arXiv* 2024, arXiv:2402.03855. [CrossRef]
14. Jermyn, A.S.; Schiefer, N.; Hubinger, E. Engineering monosemanticity in toy models. *arXiv* **2022**, arXiv:2211.09169. [CrossRef]
15. Bricken, T.; Templeton, A.; Batson, J.; Chen, B.; Jermyn, A.; Conerly, T.; Turner, N.; Anil, C.; Denison, C.; Askell, A.; et al. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. Transformer Circuits Thread. 2023. Available online: https://transformer-circuits.pub/2023/monosemantic-features/index.html (accessed on 1 January 2025).
16. Elhage, N.; Hume, T.; Olsson, C.; Schiefer, N.; Henighan, T.; Kravec, S.; Hatfield-Dodds, Z.; Lasenby, R.; Drain, D.; Chen, C.; et al. Toy Models of Superposition. Transformer Circuits Thread. 2022. Available online: https://transformer-circuits.pub/2022/toy_model/index.html (accessed on 28 April 2024).
17. Scherlis, A.; Sachan, K.; Jermyn, A.S.; Benton, J.; Shlegeris, B. Polysemanticity and capacity in neural networks. *arXiv* **2022**, arXiv:2210.01892.
18. Gurnee, W.; Nanda, N.; Pauly, M.; Harvey, K.; Troitskii, D.; Bertsimas, D. Finding Neurons in a Haystack: Case Studies with Sparse Probing. *arXiv* **2023**, arXiv:2305.01610. [CrossRef]
19. Lecomte, V.; Thaman, K.; Schaeffer, R.; Bashkansky, N.; Chow, T.; Koyejo, S. What Causes Polysemanticity? An Alternative Origin Story of Mixed Selectivity from Incidental Causes. In Proceedings of the ICLR 2024 Workshop on Representational Alignment, Vienna, Austria, 11 May 2024.
20. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *arXiv* **2020**, arXiv:2005.14165. [CrossRef]
21. Olsson, C.; Elhage, N.; Nanda, N.; Joseph, N.; DasSarma, N.; Henighan, T.; Mann, B.; Askell, A.; Bai, Y.; Chen, A.; et al. In-Context Learning and Induction Heads. Transformer Circuits Thread. 2022. Available online: https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html (accessed on 1 January 2025).
22. Sakarvadia, M.; Khan, A.; Ajith, A.; Grzenda, D.; Hudson, N.; Bauer, A.; Chard, K.; Foster, I. Attention Lens: A Tool for Mechanistically Interpreting the Attention Head Information Retrieval Mechanism. *arXiv* **2023**, arXiv:2310.16270. [CrossRef]
23. Edelman, B.L.; Edelman, E.; Goel, S.; Malach, E.; Tsilivis, N. The Evolution of Statistical Induction Heads: In-Context Learning Markov Chains. *arXiv* **2024**, arXiv:2402.11004. [CrossRef]
24. Chan, L.; Garriga-Alonso, A.; Goldwosky-Dill, N.; Greenblatt, R.; Nitishinskaya, J.; Radhakrishnan, A.; Shlegeris, B.; Thomas, N. Causal Scrubbing, A Method for Rigorously Testing Interpretability Hypotheses. AI Alignment Forum. 2022. Available online: https://www.alignmentforum.org/posts/JvZhhzycHu2Yd57RN/causal-scrubbing-a-method-for-rigorously-testing (accessed on 1 January 2025).
25. Neo, C.; Cohen, S.B.; Barez, F. Interpreting Context Look-ups in Transformers: Investigating Attention-MLP Interactions. *arXiv* **2024**, arXiv:2402.15055. [CrossRef]
26. Conmy, A.; Mavor-Parker, A.N.; Lynch, A.; Heimersheim, S.; Garriga-Alonso, A. Towards Automated Circuit Discovery for Mechanistic Interpretability. *arXiv* **2023**, arXiv:2304.14997. [CrossRef]
27. Elhage, N.; Nanda, N.; Olsson, C.; Henighan, T.; Joseph, N.; Mann, B.; Askell, A.; Bai, Y.; Chen, A.; Conerly, T.; et al. A Mathematical Framework for Transformer Circuits. Transformer Circuits Thread. 2021. Available online: https://transformer-circuits.pub/2021/framework/index.html (accessed on 1 January 2025).
28. Yu, Z.; Ananiadou, S. Locating Factual Knowledge in Large Language Models: Exploring the Residual Stream and analysing Subvalues in Vocabulary Space. *arXiv* **2024**, arXiv:2312.12141.
29. Todd, E.; Li, M.L.; Sharma, A.S.; Mueller, A.; Wallace, B.C.; Bau, D. Function Vectors in Large Language Models. *arXiv* **2024**, arXiv:2310.15213. [CrossRef]
30. Shai, A.S.; Marzen, S.E.; Teixeira, L.; Oldenziel, A.G.; Riechers, P.M. Transformers represent belief state geometry in their residual stream. *arXiv* **2024**, arXiv:2405.15943. [CrossRef]
31. Meng, K.; Bau, D.; Andonian, A.; Belinkov, Y. Locating and Editing Factual Associations in GPT. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 17359–17372.
32. Zou, A.; Phan, L.; Chen, S.; Campbell, J.; Guo, P.; Ren, R.; Pan, A.; Yin, X.; Mazeika, M.; Dombrowski, A.K.; et al. Representation Engineering: A Top-Down Approach to AI Transparency. *arXiv* **2023**, arXiv:2310.01405. [CrossRef]
33. Liu, W.; Wang, X.; Wu, M.; Li, T.; Lv, C.; Ling, Z.; Zhu, J.; Zhang, C.; Zheng, X.; Huang, X. Aligning large language models with human preferences through representation engineering. *arXiv* **2023**, arXiv:2312.15997. [CrossRef]
34. Lee, A.; Bai, X.; Pres, I.; Wattenberg, M.; Kummerfeld, J.K.; Mihalcea, R. A Mechanistic Understanding of Alignment Algorithms: A Case Study on DPO and Toxicity. *arXiv* **2024**, arXiv:2401.01967. [CrossRef]
35. Hazineh, D.S.; Zhang, Z.; Chiu, J. Linear Latent World Models in Simple Transformers: A Case Study on Othello-GPT. *arXiv* **2023**, arXiv:2310.07582. [CrossRef]

36. Marks, S.; Tegmark, M. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv* **2023**, arXiv:2310.06824. [CrossRef]
37. Burns, C.; Ye, H.; Klein, D.; Steinhardt, J. Discovering Latent Knowledge in Language Models Without Supervision. *arXiv* **2024**, arXiv:2212.03827. [CrossRef]
38. Li, K.; Hopkins, A.K.; Bau, D.; Viégas, F.; Pfister, H.; Wattenberg, M. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv* **2022**, arXiv:2210.13382.
39. Gurnee, W.; Tegmark, M. Language models represent space and time. *arXiv* **2023**, arXiv:2310.02207.
40. Ju, T.; Sun, W.; Du, W.; Yuan, X.; Ren, Z.; Liu, G. How Large Language Models Encode Context Knowledge? A Layer-Wise Probing Study. *arXiv* **2024**, arXiv:2402.16061. [CrossRef]
41. Wu, Z.; Arora, A.; Wang, Z.; Geiger, A.; Jurafsky, D.; Manning, C.D.; Potts, C. ReFT: Representation Finetuning for Language Models. *arXiv* **2024**, arXiv:2404.03592. [CrossRef]
42. Jin, M.; Yu, Q.; Huang, J.; Zeng, Q.; Wang, Z.; Hua, W.; Zhao, H.; Mei, K.; Meng, Y.; Ding, K.; et al. Exploring Concept Depth: How Large Language Models Acquire Knowledge at Different Layers? *arXiv* **2024**, arXiv:2404.07066. [CrossRef]
43. Nanda, N.; Chan, L.; Lieberum, T.; Smith, J.; Steinhardt, J. Progress measures for grokking via mechanistic interpretability. *arXiv* **2023**, arXiv:2301.05217. [CrossRef]
44. Power, A.; Burda, Y.; Edwards, H.; Babuschkin, I.; Misra, V. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv* **2022**, arXiv:2201.02177. [CrossRef]
45. Murty, S.; Sharma, P.; Andreas, J.; Manning, C.D. Grokking of Hierarchical Structure in Vanilla Transformers. *arXiv* **2023**, arXiv:2305.18741. [CrossRef]
46. Huang, Y.; Hu, S.; Han, X.; Liu, Z.; Sun, M. Unified View of Grokking, Double Descent and Emergent Abilities: A Perspective from Circuits Competition. *arXiv* **2024**, arXiv:2402.15175. [CrossRef]
47. Liu, Z.; Michaud, E.J.; Tegmark, M. Omnigrok: Grokking beyond algorithmic data. *arXiv* **2022**, arXiv:2210.01117.
48. Thilak, V.; Littwin, E.; Zhai, S.; Saremi, O.; Paiss, R.; Susskind, J.M. The Slingshot Effect: A Late-Stage Optimization Anomaly in Adaptive Gradient Methods. Transactions on Machine Learning Research. 2024. Available online: https://machinelearning.apple.com/research/slingshot-effect (accessed on 1 January 2025).
49. Furuta, H.; Minegishi, G.; Iwasawa, Y.; Matsuo, Y. Interpreting Grokked Transformers in Complex Modular Arithmetic. *arXiv* **2024**, arXiv:2402.16726. [CrossRef]
50. Chen, S.; Sheen, H.; Wang, T.; Yang, Z. Training Dynamics of Multi-Head Softmax Attention for In-Context Learning: Emergence, Convergence, and Optimality. *arXiv* **2024**, arXiv:2402.19442.
51. Liu, Z.; Kitouni, O.; Nolte, N.S.; Michaud, E.; Tegmark, M.; Williams, M. Towards understanding grokking: An effective theory of representation learning. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 34651–34663.
52. Zhu, X.; Fu, Y.; Zhou, B.; Lin, Z. Critical data size of language models from a grokking perspective. *arXiv* **2024**, arXiv:2401.10463. [CrossRef]
53. Wang, B.; Yue, X.; Su, Y.; Sun, H. Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization. *arXiv* **2024**, arXiv:2405.15071. [CrossRef]
54. Rajendran, G.; Buchholz, S.; Aragam, B.; Schölkopf, B.; Ravikumar, P. Learning Interpretable Concepts: Unifying Causal Representation Learning and Foundation Models. *arXiv* **2024**, arXiv:2402.09236. [CrossRef]
55. Tamkin, A.; Askell, A.; Lovitt, L.; Durmus, E.; Joseph, N.; Kravec, S.; Nguyen, K.; Kaplan, J.; Ganguli, D. Evaluating and mitigating discrimination in language model decisions. *arXiv* **2023**, arXiv:2312.03689. [CrossRef]
56. Doshi, D.; Das, A.; He, T.; Gromov, A. To grok or not to grok: Disentangling generalization and memorization on corrupted algorithmic datasets. *arXiv* **2023**, arXiv:2310.13061.
57. Kumar, T.; Bordelon, B.; Gershman, S.J.; Pehlevan, C. Grokking as the Transition from Lazy to Rich Training Dynamics. *arXiv* **2023**, arXiv:2310.06110.
58. Hase, P.; Bansal, M.; Kim, B.; Ghandeharioun, A. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *arXiv* **2023**, arXiv:2301.04213. [CrossRef]
59. Stolfo, A.; Belinkov, Y.; Sachan, M. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; pp. 7035–7052.
60. Hou, Y.; Li, J.; Fei, Y.; Stolfo, A.; Zhou, W.; Zeng, G.; Bosselut, A.; Sachan, M. Towards a Mechanistic Interpretation of Multi-Step Reasoning Capabilities of Language Models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; Bouamor, H., Pino, J., Bali, K., Eds.; pp. 4902–4919. [CrossRef]
61. Prakash, N.; Shaham, T.R.; Haklay, T.; Belinkov, Y.; Bau, D. Fine-Tuning Enhances Existing Mechanisms: A Case Study on Entity Tracking. *arXiv* **2024**, arXiv:2402.14811. [CrossRef]
62. Cohen, R.; Biran, E.; Yoran, O.; Globerson, A.; Geva, M. Evaluating the ripple effects of knowledge editing in language models. *arXiv* **2023**, arXiv:2307.12976. [CrossRef]

63. Xu, D.; Zhang, Z.; Zhu, Z.; Lin, Z.; Liu, Q.; Wu, X.; Xu, T.; Zhao, X.; Zheng, Y.; Chen, E. Editing Factual Knowledge and Explanatory Ability of Medical Large Language Models. *arXiv* **2024**, arXiv:2402.18099. [CrossRef]
64. Stoehr, N.; Gordon, M.; Zhang, C.; Lewis, O. Localizing Paragraph Memorization in Language Models. *arXiv* **2024**, arXiv:2403.19851. [CrossRef]
65. Sharma, A.S.; Atkinson, D.; Bau, D. Locating and Editing Factual Associations in Mamba. *arXiv* **2024**, arXiv:2404.03646. [CrossRef]
66. Yang, Y.; Duan, H.; Abbasi, A.; Lalor, J.P.; Tam, K.Y. Bias A-head? analysing Bias in Transformer-Based Language Model Attention Heads. *arXiv* **2023**, arXiv:2311.10395.
67. Jin, Z.; Cao, P.; Yuan, H.; Chen, Y.; Xu, J.; Li, H.; Jiang, X.; Liu, K.; Zhao, J. Cutting Off the Head Ends the Conflict: A Mechanism for Interpreting and Mitigating Knowledge Conflicts in Language Models. *arXiv* **2024**, arXiv:2402.18154. [CrossRef]
68. Danilevsky, M.; Qian, K.; Aharonov, R.; Katsis, Y.; Kawas, B.; Sen, P. A Survey of the State of Explainable AI for Natural Language Processing. In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Suzhou, China, 4–7 December 2020; Wong, K.F., Knight, K., Wu, H., Eds.; pp. 447–459.
69. Saeed, W.; Omlin, C. Explainable AI (XAI): A Systematic Meta-Survey of Current Challenges and Future Opportunities. *arXiv* **2021**, arXiv:2111.06420. [CrossRef]
70. Luo, H.; Specia, L. From Understanding to Utilization: A Survey on Explainability for Large Language Models. *arXiv* 2024, arXiv:2401.12874. [CrossRef]
71. Ferrando, J.; Sarti, G.; Bisazza, A.; Costa-jussà, M.R. A Primer on the Inner Workings of Transformer-based Language Models. *arXiv* 2024, arXiv:2405.00208. [CrossRef]
72. Papageorgiou, E.; Chronis, C.; Varlamis, I.; Himeur, Y. A Survey on the Use of Large Language Models (LLMs) in Fake News. *Future Internet* **2024**, *16*, 298. [CrossRef]
73. Hang, C.N.; Yu, P.D.; Morabito, R.; Tan, C.W. Large Language Models Meet Next-Generation Networking Technologies: A Review. *Future Internet* **2024**, *16*, 365. [CrossRef]
74. Krueger, D.S. Mechanistic Interpretability as Reverse Engineering (Follow-Up to "Cars and Elephants") — AI Alignment Forum — alignmentforum.org. 2022. Available online: https://www.alignmentforum.org/posts/kjRGMdRxXb9c5bWq5/mechanistic-interpretability-as-reverse-engineering-follow (accessed on 28 April 2024).
75. Olah, C.; Cammarata, N.; Schubert, L.; Goh, G.; Petrov, M.; Carter, S. Zoom In: An Introduction to Circuits. Distill. 2020. Available online: https://distill.pub/2020/circuits/zoom-in (accessed on 28 April 2024).
76. Geva, M.; Caciularu, A.; Wang, K.; Goldberg, Y. Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022; Goldberg, Y., Kozareva, Z., Zhang, Y., Eds.; pp. 30–45. [CrossRef]
77. Hanna, M.; Liu, O.; Variengien, A. How does GPT-2 compute greater-than? Interpreting mathematical abilities in a pre-trained language model. *arXiv* **2023**, arXiv:2305.00586.
78. Bereska, L.; Gavves, E. Mechanistic Interpretability for AI Safety—A Review. *arXiv* **2024**, arXiv:2404.14082.
79. Friedman, D.; Wettig, A.; Chen, D. Learning Transformer Programs. *arXiv* **2023**, arXiv:2306.01128. [CrossRef]
80. Zimmermann, R.S.; Klein, T.; Brendel, W. Scale Alone Does not Improve Mechanistic Interpretability in Vision Models. *arXiv* **2024**, arXiv:2307.05471. [CrossRef]
81. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv* **2016**, arXiv:1602.04938.
82. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4768–4777.
83. Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T.B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling Laws for Neural Language Models. *arXiv* **2020**, arXiv:2001.08361. [CrossRef]
84. Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; de Las Casas, D.; Hendricks, L.A.; Welbl, J.; Clark, A.; et al. Training Compute-Optimal Large Language Models. *arXiv* **2022**, arXiv:2203.15556. [CrossRef]
85. Das, A.; Rad, P. Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. *arXiv* **2020**, arXiv:2006.11371. [CrossRef]
86. Xu, B.; Poo, M.m. Large language models and brain-inspired general intelligence. *Natl. Sci. Rev.* **2023**, *10*, nwad267. [CrossRef]
87. Sharkey, L.; Braun, D.; beren. Interim Research Report Taking Features out of Superposition with Sparse Autoencoders. 2022. Available online: https://www.lesswrong.com/posts/z6QQJbtpkEAX3Aojj/interim-research-report-taking-features-out-of-superposition (accessed on 23 January 2024).
88. McDougall, C.; Conmy, A.; Rushing, C.; McGrath, T.; Nanda, N. Copy suppression: Comprehensively understanding an attention head. *arXiv* **2023**, arXiv:2310.04625. [CrossRef]
89. Cammarata, N.; Goh, G.; Carter, S.; Schubert, L.; Petrov, M.; Olah, C. Curve Detectors. Distill. 2020. Available online: https://distill.pub/2020/circuits/curve-detectors (accessed on 28 April 2024).

90. Schubert, L.; Voss, C.; Cammarata, N.; Goh, G.; Olah, C. High-Low Frequency Detectors. *Distill*. 2021. Available online: https://distill.pub/2020/circuits/frequency-edges (accessed on 28 April 2024).
91. Olah, C.; Cammarata, N.; Voss, C.; Schubert, L.; Goh, G. Naturally Occurring Equivariance in Neural Networks. *Distill*. 2020. Available online: https://distill.pub/2020/circuits/equivariance (accessed on 28 April 2024).
92. Nanda, N.; Lee, A.; Wattenberg, M. Emergent linear representations in world models of self-supervised sequence models. *arXiv* **2023**, arXiv:2309.00941. [CrossRef]
93. Li, M.; Davies, X.; Nadeau, M. Circuit Breaking: Removing Model behaviours with Targeted Ablation. *arXiv* **2024**, arXiv:2309.05973.
94. Li, K.; Patel, O.; Viégas, F.; Pfister, H.; Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 41451–41530.
95. Azaria, A.; Mitchell, T. The internal state of an llm knows when its lying. *arXiv* **2023**, arXiv:2304.13734. [CrossRef]
96. CH-Wang, S.; Van Durme, B.; Eisner, J.; Kedzie, C. Do Androids Know They're Only Dreaming of Electric Sheep? *arXiv* **2023**, arXiv:2312.17249.
97. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. Emergent Abilities of Large Language Models. *arXiv* **2022**, arXiv:2206.07682. [CrossRef]
98. Thilak, V.; Littwin, E.; Zhai, S.; Saremi, O.; Paiss, R.; Susskind, J.M. The Slingshot Mechanism: An Empirical Study of Adaptive Optimizers and the *Grokking Phenomenon*. In Proceedings of the Has it Trained Yet? NeurIPS 2022 Workshop, New Orleans, LA, USA, 2 December 2022.
99. Bhaskar, A.; Friedman, D.; Chen, D. The Heuristic Core: Understanding Subnetwork Generalization in Pretrained Language Models. *arXiv* **2024**, arXiv:2403.03942. [CrossRef]
100. Bushnaq, L.; Mendel, J.; Heimersheim, S.; Braun, D.; Goldowsky-Dill, N.; Hänni, K.; Wu, C.; Hobbhahn, M. Using Degeneracy in the Loss Landscape for Mechanistic Interpretability. *arXiv* **2024**, arXiv:2405.10927. [CrossRef]
101. Golechha, S. Progress Measures for Grokking on Real-world Datasets. *arXiv* **2024**, arXiv:2405.12755. [CrossRef]
102. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
103. Lyu, K.; Jin, J.; Li, Z.; Du, S.S.; Lee, J.D.; Hu, W. Dichotomy of Early and Late Phase Implicit Biases Can Provably Induce Grokking. *arXiv* **2023**, arXiv:2311.18817.
104. Mohamadi, M.A.; Li, Z.; Wu, L.; Sutherland, D. Grokking modular arithmetic can be explained by margin maximization. In Proceedings of the NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning, New Orleans, LA, USA, 16 December 2023.
105. Merrill, W.; Tsilivis, N.; Shukla, A. A Tale of Two Circuits: Grokking as Competition of Sparse and Dense Subnetworks. *arXiv* **2023**, arXiv:2303.11873. [CrossRef]
106. Nakkiran, P.; Kaplun, G.; Bansal, Y.; Yang, T.; Barak, B.; Sutskever, I. Deep double descent: Where bigger models and more data hurt. *J. Stat. Mech. Theory Exp.* **2021**, *2021*, 124003. [CrossRef]
107. Davies, X.; Langosco, L.; Krueger, D. Unifying Grokking and Double Descent. *arXiv* **2023**, arXiv:2303.06173. [CrossRef]
108. Chen, W.; Song, J.; Ren, P.; Subramanian, S.; Morozov, D.; Mahoney, M.W. Data-Efficient Operator Learning via Unsupervised Pretraining and In-Context Learning. *arXiv* **2024**, arXiv:2402.15734.
109. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *arXiv* **2021**, arXiv:2106.09685.
110. Wu, M.; Liu, W.; Wang, X.; Li, T.; Lv, C.; Ling, Z.; Zhu, J.; Zhang, C.; Zheng, X.; Huang, X. Advancing Parameter Efficiency in Fine-tuning via Representation Editing. *arXiv* **2024**, arXiv:2402.15179. [CrossRef]
111. Held, W.; Yang, D. Shapley Head Pruning: Identifying and Removing Interference in Multilingual Transformers. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Dubrovnik, Croatia, 2–6 May 2023; Vlachos, A., Augenstein, I., Eds.; pp. 2416–2427. [CrossRef]
112. Jain, S.; Kirk, R.; Lubana, E.S.; Dick, R.P.; Tanaka, H.; Grefenstette, E.; Rocktäschel, T.; Krueger, D.S. Mechanistically analysing the effects of fine-tuning on procedurally defined tasks. *arXiv* **2023**, arXiv:2311.12786.
113. Turner, A.; Thiergart, L.; Udell, D.; Leech, G.; Mini, U.; MacDiarmid, M. Activation addition: Steering language models without optimization. *arXiv* **2023**, arXiv:2308.10248. [CrossRef]
114. Geiger, A.; Wu, Z.; Potts, C.; Icard, T.; Goodman, N. Finding alignments between interpretable causal variables and distributed neural representations. In Proceedings of the Causal Learning and Reasoning, PMLR, Los Angeles, CA, USA, 1–3 April 2024; pp. 160–187.
115. Cao, N.D.; Aziz, W.; Titov, I. Editing Factual Knowledge in Language Models. *arXiv* **2021**, arXiv:2104.08164. [CrossRef]
116. Hernandez, E.; Li, B.Z.; Andreas, J. Inspecting and Editing Knowledge Representations in Language Models. *arXiv* **2023**, arXiv:2304.00740. [CrossRef]

117. Zhang, F.; Nanda, N. Towards Best Practices of Activation Patching in Language Models: Metrics and Methods. *arXiv* **2024**, arXiv:2309.16042. [CrossRef]
118. Campbell, J.; Ren, R.; Guo, P. Localizing Lying in Llama: Understanding Instructed Dishonesty on True-False Questions Through Prompting, Probing, and Patching. *arXiv* **2023**, arXiv:2311.15131. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.