101168951 – ARMADA
HORIZON – MSCA – 2023 – DN
State of the Art Report

A Survey of New Frontiers in Explainability for Large Language Models: From Interpretation to Intervention

State of the Art Report on Explanation Frameworks and Explainability Techniques for Large Language Models

Tommaso Dolci* · **Dina Fettache*** · **Vittorio La Ferla*** · **Amanda Le*** · **Sihem Amer-Yahia** · **Jasmina Bogojeska** · **Katja Hose** · **Noha Ibrahim** · **Silviu Maniu** · **Katya Mirylenka** · **Maxime Peyrard**

Last updated: October 23, 2026

Abstract Large language models (LLMs) achieve strong performance across a number of generative AI tasks, yet their internal decision-making remains largely opaque, limiting trust and safe deployment in high-stakes domains such as healthcare, finance, and law. While Explainable AI (XAI) has produced diverse methods for interpreting traditional AI models, the scale and emergent behaviors of LLMs pose a challenge to conventional explainability approaches. Moreover, explanations are increasingly being exploited not only for improving human understanding, but also for enhancing model performance and reliability. This survey re-examines explainability methods for LLMs through the lens of interpretability of the model and model improvement, arguing that the field is evolving towards both passive interpretation and active intervention, i.e., explanations diagnose failures, guide model updates, and improve reasoning. To structure this novel perspective, we introduce a taxonomy distinguishing *explainability of the prediction* (i.e., uncovering mechanisms behind specific outputs) and *explainability for the predictor* (i.e., characterizing broader model behavior to improve the model or system). We further review domain-specific applications and key open challenges about XAI for LLMs, including scalability, faithfulness of explanations, and causal validation.

Keywords Large Language Model · Explainable AI · Interpretability · Survey

Sihem Amer-Yahia, Noha Ibrahim, Dina Fettache, Amanda Le, Silviu Maniu, Maxime Peyrard
CNRS, Univ. Grenoble Alpes, Grenoble, France
E-mail: {firstname.lastname}@univ-grenoble-alpes.fr

Tommaso Dolci, Katja Hose, Katya Mirylenka
TU Wien, Vienna, Austria
E-mail: {firstname.lastname}@tuwien.ac.at

Jasmina Bogojeska, Vittorio La Ferla
Zurich University of Applied Sciences, Winterthur, Switzerland
E-mail: {lafe,bogo}@zhaw.ch

* These authors contributed equally.

1 Introduction

Large language models (LLMs) have emerged as one of the most groundbreaking developments in AI, demonstrating remarkable capabilities across a number of tasks, including language generation [19], reasoning [176], code synthesis [76], semantic parsing and Text-to-SQL [60,78], and knowledge retrieval [218]. Yet, despite their impressive performance, LLMs are not intrinsically explainable [7,211]. Therefore, their predictions, reasoning processes, internal representations, and failure modes remain largely opaque [40], limiting trust, accountability, and safe deployment, especially in high-stakes domains such as healthcare, law and finance.

Challenges. The field of Explainable Artificial Intelligence (XAI) has produced a diverse set of methods for understanding traditional machine learning models [40,136]. However, this body of work was designed primarily for discriminative prediction tasks over structured inputs. On the other hand, current LLMs have billions of parameters and densely distributed representations, features that violate the assumptions underlying classical attribution methods. For instance, explainability approaches such as LIME [138] and SHAP [100] require repeated model evaluations over perturbed inputs, making them impractical for large-scale LLMs.

Beyond scalability, modern LLMs exhibit *neuron polysemanticity* [14], i.e., the property whereby a single neuron activates for multiple unrelated concepts. Furthermore, due to the auto-regressive nature of generative LLMs, the output depends heavily on the input context and the previously generated tokens. These characteristics make explaining LLM behavior more challenging than interpreting a single prediction, and it remains uncertain how to reliably assess whether traditional explanations genuinely reflect the model's underlying reasoning [164].

As the role of AI systems expands from input classification to open-ended output generation, reasoning over long contexts, and acting as autonomous agents, the question of what an explanation should achieve changes fundamentally. We argue that explanations should be treated both as post-hoc artifacts to help humans read a single output and as instruments for diagnosing and improving the underlying AI system itself. Therefore, this survey organizes explanations for LLMs into passive interpretation (explainability of the prediction) and active intervention (explainability for the prediction).

Contributions. Guided by this perspective, we organize this survey around four main contributions:

C1. We re-frame explainability for LLMs as an evolution from traditional XAI techniques towards both passive interpretation and active intervention.
C2. We categorize existing XAI methods for LLMs through a taxonomy centered on explainability-driven improvement, distinguishing between *explainability of the prediction* and *explainability for the predictor*.
C3. We survey concrete advances in LLM explainability through mechanistic intervention, reasoning improvement, external knowledge grounding, retroactive learning, and their impact on high-stakes domains (e.g., healthcare, law, finance).
C4. We examine key open challenges and research directions in XAI for LLMs, highlighting issues such as scalability, explanation faithfulness, and the causal validation of explanations.

Outline. The rest of the paper is organized as follows. Section 2 introduces and discusses a novel taxonomy of XAI methods. Section 3 focuses on explainability of the prediction, including mechanistic and self-explainability techniques to passively interpret model outputs. Section 4 focuses on explainability for the predictor, including evidence-based approaches and retroactive explainability to actively enhance the performance and transparency of LLMs. Section 5 discusses the application of XAI techniques in high-stakes domains. Section 6 presents an overview of open challenges and future directions in the research area. Finally, Section 7 summarizes the main findings and concludes the paper.

Related Surveys. Over the years, numerous surveys have examined XAI for traditional ML, providing comprehensive overviews of existing approaches [42], identifying open challenges [35], and discussing the role of explainability in responsible AI practices [12]. More recently, attention has shifted towards explainability for LLMs. Existing surveys have investigated XAI techniques across different stages of the LLM lifecycle [211] and reviewed the practical utilization of XAI techniques [102]. Motivated by recent calls to rethink the notion of explainability in the era of LLMs [149], our survey organizes the literature distinguishing between approaches for *passively* describing the model output and approaches that *actively* enhance LLMs through interventions designed to improve performance, transparency, and reliability.

2 Taxonomy

Explainability for LLMs extends beyond the objectives traditionally associated with XAI. Classical post-hoc methods such as LIME [138] and SHAP [100] were primarily developed for traditional machine learning models. However, LLMs generate open-ended sequences, reason over long contexts, and increasingly operate as components of larger systems. Consequently, explaining a single prediction is only one of several roles that explainability can play. Explanations can also expose internal mechanisms, diagnose failures, and provide signals that are fed back into the system to modify its subsequent behavior.

Therefore, we organize the emerging landscape of XAI for LLMs according to the purpose of the explanation, distinguishing between *explainability of the prediction* and *explainability for the predictor*. Figure 1 illustrates the taxonomy of XAI approaches for LLMs.

Explainability of the prediction asks why a particular output was produced. This category includes mechanistic approaches that identify and causally validate internal computations underlying model behavior, as well as reasoning-based approaches that expose intermediate natural-language reasoning traces. In both cases, the explanation is primarily used to make a particular model behavior more understandable and auditable.

On the other hand, *explainability for the predictor* considers explanatory information as a resource for improving the model or the surrounding system. In particular, evidence-based approaches ground inference in explicit external knowledge, e.g., knowledge graphs or

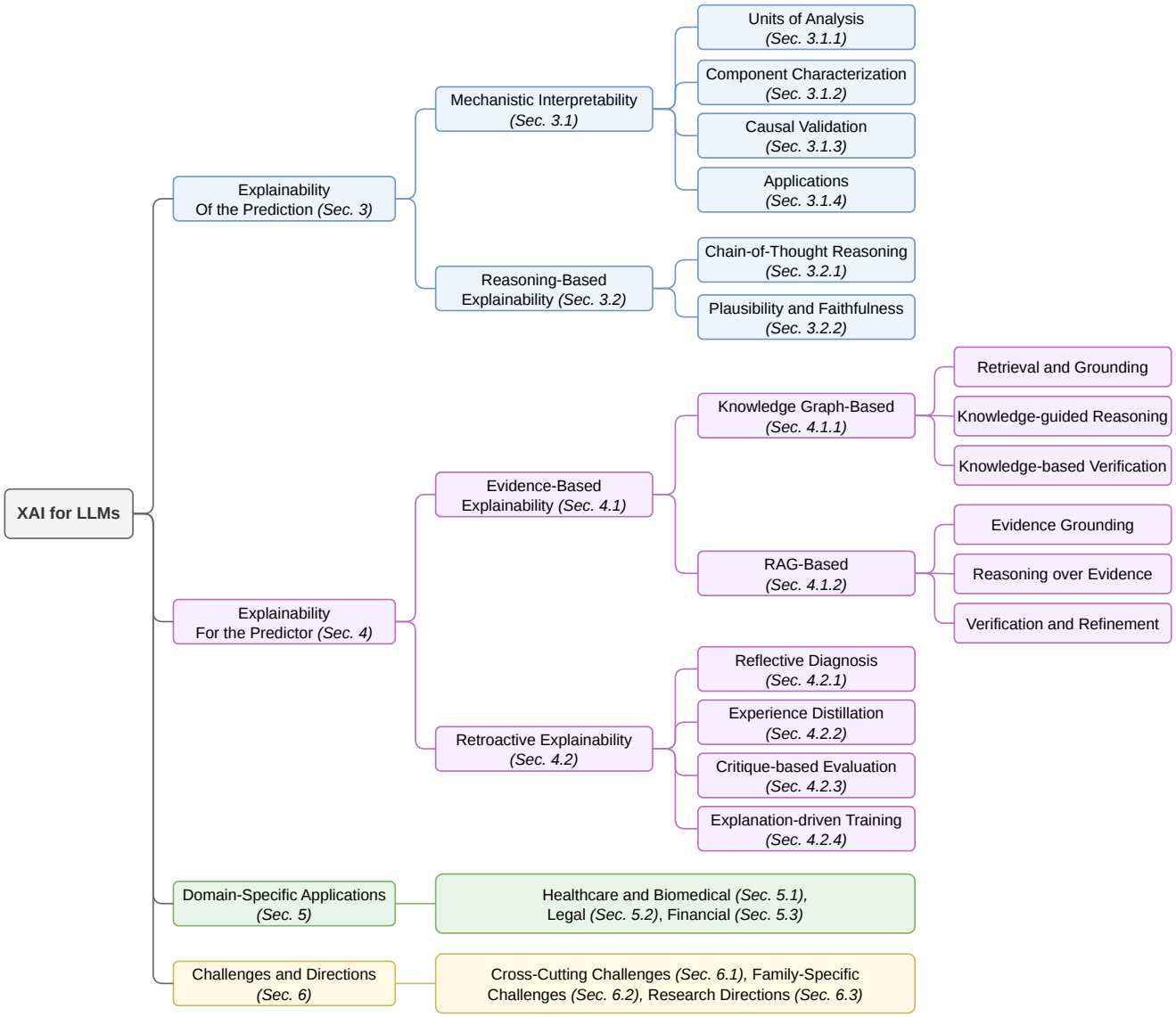


Fig. 1 New frontiers of explainability for LLMs, organized around explainability of the prediction and explainability for the predictor, together with domain-specific applications and open challenges.

retrieved documents, making evidence available both for inspection and for generation. Retroactive approaches generate explanations of previous outcomes, such as critiques, diagnoses, or distilled strategies, and feed these explanations back into the system to guide behavior. Therefore, explainability becomes a signal to influence what the model will do next.

> **Key Takeaway.** Explanations can help us understand a single prediction or become signals for improving what the predictor will do next.

The distinction is functional rather than absolute. The same explanatory artifact may serve both purposes: a mechanistic explanation can reveal why an output occurred and provide a target for model intervention, while retrieved evidence can simultaneously justify an answer and improve its factual reliability. Our taxonomy therefore considers both how explanations help us understand model behavior and how they can be used to improve it. The remainder of the survey examines these methods, their applications in high-stakes domains, and the shared challenges of faithfulness, evaluation, scalability, and human usability. The survey adopts a narrative approach, focusing on relevant contributions that illustrate the main methodological families of XAI for LLMs. The selection is guided by the conceptual scope of the survey rather than by a systematic search protocol.

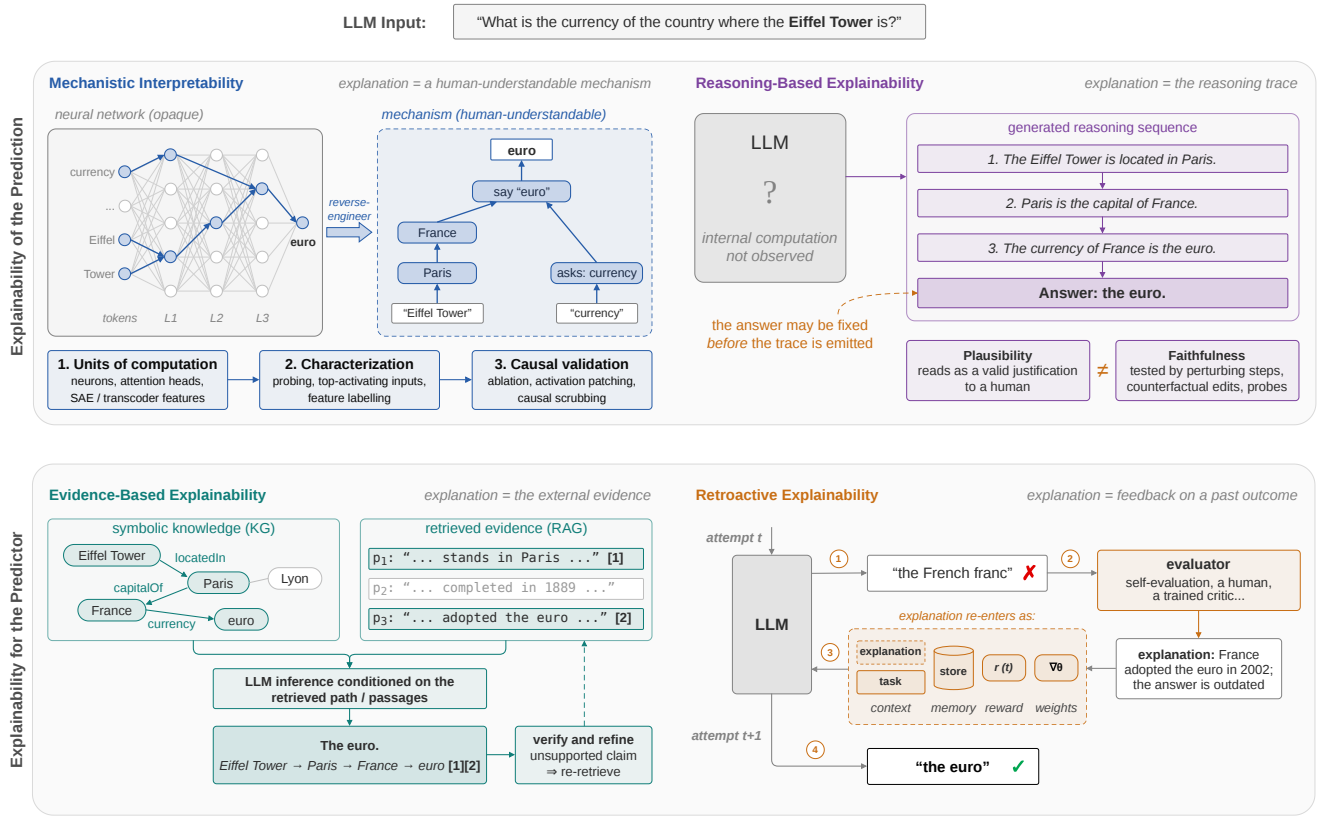


Fig. 2 Overview of new approaches in explainability for LLMs.

3 Explainability of the Prediction

This section covers explanation of individual model outputs: accounts of why a specific prediction was produced. We distinguish two families by where the account comes from. Mechanistic interpretability derives it from the computation itself, decomposing the model into internal components and testing their causal role. Reasoning-based explainability derives it from the model's own generated reasoning traces.

3.1 Mechanistic Interpretability

Mechanistic interpretability of LLMs aims to understand neural networks by reverse-engineering their internal computations into interpretable components, mechanisms, and representations [14, 120, 205]. It does so by decomposing a model into smaller internal components and analyzing how these components and their interactions contribute to model behavior [132]. This reverse-engineering process can be organized into three stages: identifying the units of computation, characterizing their functional roles, and validating these characterizations [145].

LLMs are built from stacked transformer blocks, which expose units of analysis at several levels of granularity: individual neurons, attention heads, and the residual stream that components read from and write to across layers [49]. Which unit to pick is not obvious. Neurons are the natural architectural choice, but language models may represent more features than they have dimensions and therefore store them in superposition [44]. Individual neurons can therefore be polysemantic, responding to multiple unrelated concepts [14]. The choice of unit affects both the granularity of the resulting explanation and the cost of obtaining it [116]. The following sections present representative methods at each stage. Table 1 summarizes this progression and representative methods used at each stage.

As an illustrative example of mechanistic interpretability, Wang et al. [169] investigated how GPT-2 small performs indirect object identification. Given the sentence "When Mary and John went to the store, John gave a drink to", the model predicts "Mary". Rather than merely observing this prediction, the goal is to understand the internal computations that produce it. The authors proposed a circuit of attention heads intended to explain this behavior through three computational steps: identifying the names in the sentence (Mary, John,

Table 1 Overview of mechanistic interpretability methods for LLMs.

Stage	Main Question	Representative Methods
Units of Analysis	What are the components?	Architectural components (neurons, attention heads, layers) [116]; sparse autoencoders [33]
Component Characterization	What does each component represent or compute?	Probing [1], automated LLM feature interpretation [17]
Causal Validation	Do the components causally contribute to the behavior, and how do they interact?	Ablation [110], activation patching [119], causal tracing [109], causal abstraction [52, 118], circuit tracing [3]

John), detecting the duplicated name (John), and selecting the remaining name (Mary). Such proposed explanations require causal validation, for example, by intervening on internal components and examining whether the resulting changes in model behavior agree with the proposed mechanism. More broadly, mechanistic interpretability seeks to develop such explanations for increasingly complex behaviors in LLMs.

3.1.1 Units of Analysis

The first step of mechanistic interpretability is deciding what to decompose the model into. Architectural components are the obvious candidates: neurons, attention heads, and layers are finite, enumerable, and require no additional machinery to inspect [116]. However, these units of analysis have different limitations. Layers and submodules are easy to enumerate but too coarse to interpret directly, while neurons are fine-grained yet polysemantic, so inspecting one in isolation rarely yields a single concept [116, 145].

Sparse autoencoders (SAEs) were proposed to recover more interpretable and monosemantic feature-level units [33]. An SAE maps activations into a higher-dimensional but sparsely activating latent space, with the aim that each latent responds to one interpretable feature [33]. In the context of LLMs, SAEs are typically trained on internal activations collected from extracted from transformer components, such as the residual stream, multilayer perceptron (MLP) outputs, or attention outputs. Subsequent work refined the sparsity mechanism: k-sparse autoencoders constrain the number of active latents directly, which simplifies tuning and improves the trade-off between reconstruction quality and sparsity [50]. SAEs have also been scaled to large production models such as Claude 3 Sonnet, where they have recovered recognizable features, including the "Golden Gate Bridge" feature [161].

Despite these advances, SAEs still have significant limitations as a basis for model decomposition. Trained SAEs can contain dead latents that rarely or never activate [50], and the learned dictionaries are not necessarily canonical [88]. In particular, the features recovered can depend on the SAE's dictionary size: larger SAEs may learn more fine-grained or entirely novel features, while some latents can themselves decompose into combinations of features found at other scales [88]. As a result, SAEs do not provide a unique or canonical decomposition of the model, leaving open the more fundamental question of what the appropriate units of mechanistic analysis should be.

> **Key Takeaway.** Decoding a property from a model's internal representations shows only that the information is present, not that the model relies on it to produce its output.

3.1.2 Component Characterization

Once a unit of analysis has been selected, the next step is to characterize what information it represents or what role it may play in the model's computation. A common approach is probing [1], where an auxiliary classifier, typically a linear classifier, is trained on hidden activations to determine whether a particular property is linearly decodable from the model's internal representations. Probing has been used to identify a wide range of information, including signals associated with hallucinations [22] and representations of real-world spatial and temporal structure [59].

Another approach characterizes components through the inputs that most strongly activate them [17]. By inspecting highly activating examples, researchers can infer the concepts or patterns to which a component responds. This strategy can also be applied to SAE features by examining the tokens or contexts associated with strong latent activations. As SAEs can contain millions of features, manually characterizing each feature from its activation patterns is impractical. Automated interpretation methods therefore use LLMs to generate natural-language descriptions of feature behavior and evaluate how well these descriptions predict activations on other inputs [17].

Although these methods make internal representations easier to interpret, they mainly reveal correlations.

Being able to decode information from a representation, or finding that a component activates for a particular concept, does not show that the model actually uses that component to produce the behavior [13]. Recent work also shows that interpretability methods can produce plausible-looking explanations even in randomized models, further motivating the need for causal validation [117].

3.1.3 Causal Validation

Component characterization can identify what information is associated with an internal representation, but establishing whether that representation actually contributes to a model's behavior requires interventions. Causal validation methods therefore modify internal activations and measure how the model's output changes. A basic intervention is ablation, where an internal component, such as a neuron or attention head, is removed or replaced and the resulting change in model behavior is measured [110]. A substantial change following ablation provides evidence that the component contributes to the target behavior, although redundancy and interactions between components can complicate this interpretation.

Activation patching provides a more targeted intervention by replacing an activation produced on one input with the same internal activation produced on another input and measuring the resulting effect on model behavior [119]. Related approaches use such interventions to localize where particular information is processed. Causal tracing, for example, corrupts information in the model's input and then selectively restores internal activations to their values from a clean run, i.e., a normal forward pass on the original, uncorrupted input. If restoring a particular activation recovers the original prediction, this provides evidence that the corresponding internal state is causally important for producing that prediction [109]. This approach has been used to localize factual associations in language models [109] and has subsequently been extended to analyze in-context grounding and distinguish grounding from internal parametric recall [115].

More structured methods test causal hypotheses involving multiple components. Causal scrubbing starts from a proposed explanation of how a model performs a task and uses resampling interventions to test whether the model preserves the behavior implied by that explanation [23]. Rather than asking only whether an individual component influences the output, the goal is to test whether a hypothesized computational structure captures the dependencies responsible for the behavior. More generally, causal abstraction formalizes this idea by asking whether an interpretable high-level causal model faithfully captures the causal structure of the lower-level computation performed by a neural network [52, 118].

Ultimately, the goal of mechanistic interpretability is not only to identify individual components that influence a behavior, but to recover the mechanisms through which these components interact to produce it. These interactions can be represented as computational graphs, where nodes correspond to model components, such as attention heads or MLPs, and directed edges represent how the outputs of one component contribute to the inputs of another [62]. Circuit discovery approaches this goal by identifying subsets of a model's computational graph that implement a target behavior and characterizing the roles of their components [62]. Recent work on circuit tracing extends this idea to larger language models by constructing attribution graphs that trace computational pathways from input features through intermediate representations to model outputs [3]. However, substantial challenges remain in obtaining comprehensive mechanistic explanations of large language models [3].

3.1.4 Applications of Mechanistic Interpretability

Mechanistic interpretability can move beyond explanation by using identified components and mechanisms as targets for intervention. Zhang et al. [205] describe this as a "Locate, Steer, and Improve" pipeline, highlighting applications across alignment, capability, and efficiency. Mechanistic insights can, for example, provide targets for mitigating undesirable behaviors, modifying specific model capabilities, or identifying redundant computation that may be pruned for greater efficiency [205]. In this way, mechanistic interpretability can provide not only explanations of model behavior, but also actionable knowledge for improving it. However, we classify it under *explainability of the prediction* because its primary objective is to identify and understand the internal mechanisms responsible for model behavior.

3.2 Reasoning-Based Explainability

Reasoning-based interpretability of LLMs relies on chain-of-thought (CoT) prompting, which directly leverages the reasoning traces generated by LLMs during inference rather than relying on external explainability methods. CoT elicits intermediate natural-language reasoning steps that accompany a model's final answer, promising to boost the transparency of the internal LLM process to otherwise opaque neural computation. CoT has rapidly gained popularity not only as a performance-boosting technique but also as a perceived window into model reasoning: a recent analysis by [11] estimates that 25%

of arXiv papers between April 2024 and June 2025 that implement CoT mechanisms also treat them as an interpretability method. While CoT prompting has become central to the explainability of LLMs, its explanatory value remains an open question. The key issue is whether CoT traces genuinely explain how the model arrived at its prediction or merely provide a plausible post-hoc justification. In the remainder of this section, we examine the explanatory value of CoT by discussing both the arguments supporting its use as an explanation and the limitations that challenge its faithfulness.

3.2.1 Chain-of-Thought Reasoning

A *chain of thought* is a sequence of intermediate reasoning steps, expressed in natural language, that precedes and supports the generation of a final answer. The idea was popularized by Wei et al. [176], who showed that providing a few manually written reasoning demonstrations (i.e., few-shot CoT) can improve the LLM performance on arithmetic, commonsense, and symbolic reasoning tasks. Kojima et al. [80] demonstrated that the simple prompt "Let's think step by step" (i.e., zero-shot CoT) can elicit comparable behavior without providing any hand-crafted example. To further improve the robustness of CoT prompting, self-consistency decoding samples multiple reasoning paths and selects the most frequently occurring answer through majority voting, yielding a more robust inference strategy [172]. This consistency-based paradigm can be extended into formal uncertainty quantification (UQ) to explicitly measure the confidence and reliability of the generated reasoning, particularly in complex generation, Question Answering and Text-to-SQL tasks [15, 16, 186]. The rapidly growing family of CoT methods has given rise to numerous variations of prompting and reasoning strategies [31], including program-of-thoughts [27], which delegates intermediate computations to external program execution, tree-of-thoughts [193], which explores multiple reasoning paths through structured tree search, and graph-of-thoughts [195], which generalizes reasoning trajectories into graph-based structures.

Beyond the canonical prompting paradigms, subsequent work has refined the understanding of the factors underlying the effectiveness of CoT. For instance, the importance of both semantic content and structural patterns in CoT-based few-shot prompting [106] suggests that models exploit not only the textual reasoning steps but also formal characteristics of the provided demonstrations. Perturbation-based analyses of CoT reasoning [67] revealed that models retain 80–90% of their baseline performance even when intermediate reasoning steps are rendered logically invalid, indicating that factors such as the ordering of steps and their relevance to the query are more important than the logical validity of each individual step. The continued evolution of CoT is exemplified by Chain-of-Knowledge prompting [167], which augments reasoning traces with structured knowledge triples and extends the paradigm beyond purely natural-language reasoning. Recently, reasoning-oriented LLMs have emerged as a new direction for the CoT paradigm by integrating extended reasoning processes into model training rather than relying solely on prompting strategies. Models such as OpenAI o1 [122] and DeepSeek-R1 [58] leverage reinforcement learning to encourage the generation of longer reasoning trajectories and more robust behaviors such as self-verification and reflection.

3.2.2 Explanations, Plausibility, and Faithfulness

Despite its origins as a reasoning enabler, CoT possesses several properties that make it intuitively appealing as an explanation: (i) it decomposes complex problems into human-readable sub-steps, making intermediate assumptions explicit and exposing intermediate errors that would otherwise remain hidden; (ii) the resulting traces serve as human-readable explanations that accompany the final prediction, increasing users' perceived transparency and trust; (iii) when the reasoning process leads to an error, the step at which the failure occurs is more easily identifiable, enabling targeted debugging.

However, to properly evaluate CoT as an explainability tool, it is necessary to distinguish between plausible explanations and faithful explanations [72, 94]. Plausible explanations provide a coherent and convincing account of a model's prediction, but they may not reflect the actual rationale underlying the decision process. In contrast, faithful explanations accurately capture the rationale that contributed to the model's output, reflecting the relevant factors or computations involved in producing the prediction. Table 2 provides an overview of plausible and faithful explanations.

Barez et al. [11] argue that CoT reasoning is optimized for plausibility, not faithfulness: the decoding objective rewards tokens likely under the preceding context, not tokens that accurately describe the model's internal computations. Early empirical evidence for this concern comes from perturbation studies of reasoning traces in few-shot settings [164]. By introducing systematic biases into demonstrations – such as making the correct answer always correspond to the first option in multiple-choice settings, or including a suggested answer in the prompt – this study shows that LLM reasoning steps can be substantially influenced by surface-level cues. Crucially, the generated CoT traces often fail to acknowl-

Table 2 Comparison of plausible and faithful explanations.

Type	Primary Goal	Causal Criterion	Typical Test	Risks
Plausible Explanation	Provide a human-readable justification	No causal commitment: plausibility relies on the model's next-token distribution	Paraphrase/mistake-edit perturbations [177], bias-injection tests [11]; answer-first detection [90]	Trust of fluent but unfaithful narratives [147]; masking biases [11]
Faithful Explanation	Reveal the actual decision process	Interventions that enforce or violate stated steps change the model prediction	Counterfactual input edits [177]; reasoning perturbation [86]; task decomposition [131]	Constraining can reduce performance [131]; partial adherence to internal operations [28]

edge these biases; instead, LLMs construct seemingly coherent justifications for biased predictions, revealing a disconnection between the expressed reasoning and the factors influencing the output. A related issue emerged in the attention-as-explanation debate, where attention weights were shown not to necessarily provide faithful explanations of model decisions [143], highlighting a broader challenge in interpreting model-generated artifacts as explanations.

Analyses of whether CoT actively guides reasoning or instead functions as a mere post-hoc rationalization reveal settings in which the final answer appears to be determined before the reasoning trace is fully generated, with the CoT serving primarily as a justificatory role [90]. This concern extends to modern reasoning-oriented LLMs, where evidence suggests that *models do not always say what they think*: verbalized reasoning can systematically diverge from internal computation, even in models explicitly trained to perform step-by-step reasoning [28]. Further investigations challenge the assumption that longer and more detailed reasoning traces necessarily reflect deeper or more genuine reasoning. Studies of models across different levels of problem complexity reveal the *illusion of thinking*: elaborate CoT traces may superficially resemble deliberation while failing to scale meaningfully with task difficulty, suggesting that verbosity is not a reliable indicator of faithfulness [147]. To address these challenges, empirical frameworks for evaluating the faithfulness of CoT explanations have been proposed, providing structured tests that move beyond surface-level plausibility and enable more systematic auditing of reasoning traces [25].

On the positive side, prompt design has been shown to have a direct influence on reasoning faithfulness: question decomposition, which breaks a complex query into a sequence of simpler sub-questions, produces reasoning traces that are measurably more faithful to the model's internal computations than standard CoT [131]. This suggests that faithfulness is not fixed but can be improved through careful prompting and task decomposition. Similarly, Wei et al. [178] directly study the interpretability qualities of reasoning explanations produced by prompting LLMs according to multiple prompting strategies and model sizes. They evaluate three desirable dimensions – robustness, faithfulness, and utility – by measuring on CoT explanations the effect of systematic perturbations applied to the input question, including paraphrasing, mistake editing, and counterfactual reasoning. Their results reveal that more advanced CoT strategies can improve utility and reduce unfaithfulness, although their effects vary across different dimensions of interpretability.

Overall, the current literature suggests that CoT should generally be viewed as an interpretable reasoning trace rather than a guaranteed faithful explanation of model decision-making. Therefore, while reasoning traces are valuable for improving human understanding and enabling error diagnosis, equating them with genuine explanations of the model's internal process is, currently, unsupported by sufficient empirical evidence.

> **Key Takeaway.** Reasoning traces should be treated as useful artifacts for human understanding and debugging, not necessarily as faithful evidence of the model's internal functioning.

4 Explainability for the Predictor

This section covers explanation directed at the model or system itself: accounts that serve not only to justify the LLM output to an external audience, but also to make the predictor more transparent, correctable, or capable. We distinguish two families based on the type of intervention. Evidence-based explainability achieves transparency by grounding the model's inference in external knowledge sources such as knowledge graphs or retrieved documents. Retroactive explainability intervenes after the predictor has completed the task, retrospectively generating explanations of prior outcomes and feeding them back into the system to shape the subsequent behavior of the model.

4.1 Evidence-Based Explainability

Among approaches for explainability for the predictor, evidence-based methods offer a different perspective from conventional post-hoc explainability techniques. Rather than analyzing the behavior of the LLM to infer the reasoning behind its predictions, these approaches redesign the predictor itself by integrating the expressive learning capabilities of neural networks with the explicit reasoning mechanisms of symbolic AI. Through the incorporation of symbolic knowledge and reasoning into the inference process, evidence-based methods make the model's decision-making more transparent, grounded, and verifiable. Consequently, explainability is achieved not only by interpreting model outputs after prediction, but by embedding interpretable reasoning directly into the system architecture. Evidence-based methods can be categorized into *knowledge graph-based methods*, which leverage structured symbolic knowledge to support, expose, and validate the reasoning process, and *RAG-based methods*, which augment model predictions with externally retrieved web or textual documents to improve transparency and factual reliability.

4.1.1 Knowledge Graph-Based Explanations

A knowledge graph (KG) is a structured representation of knowledge that organizes real-world entities and their relationships in a graph-based format [65]. Facts are represented as triples in the form *(head entity, relation, tail entity)*, enabling information to be stored in a machine-readable form and supporting knowledge retrieval, reasoning, and inference. Lately, KGs have been increasingly used as a source of evidence for making LLMs more transparent, factual, and auditable [124,125]. Their value for explainability comes from the fact that they represent knowledge explicitly: entities, relations, types, and constraints are stored in a form that can be inspected and queried independently from the neural model. This property makes KGs qualitatively different from the implicit knowledge stored in LLM parameters: when an LLM prediction can be linked to a KG triple, a graph path, or an ontological constraint, the resulting explanation is more structured and more verifiable than a free-form natural-language rationale. Large-scale resources such as Wikidata [165], DBpedia [8], and ConceptNet [154] provide broad factual coverage, while domain-specific KGs and ontologies support reasoning in areas such as medicine [18] and geography [54].

Most existing surveys of KG–LLM integration organize methods according to architectural stages, e.g., whether the KG is used before inference, during inference, or after inference [38,87]. This organization is consistent with broader roadmaps for synergistic KG–LLM systems [39, 125]. However, in the context of interpretability, we classify methods according to the supporting role of the KG: provide evidence, guide multi-hop reasoning, or validate generated claims [101]. We therefore organize KG-based explainability methods into three functional categories: *knowledge retrieval and grounding*, *knowledge-guided reasoning*, and *knowledge-based verification*, as described in Table 3. These categories are not mutually exclusive, e.g., Think-on-Graph [156] retrieves KG evidence and reasons over it. Similarly, StructGPT [75] can be viewed as supporting both grounding and tool-mediated reasoning over structured knowledge. The taxonomy therefore identifies the primary explainability function of each method.

Knowledge Retrieval and Grounding The first role of KGs is to ground LLM predictions in explicit evidence. In these methods, the KG acts as an external memory: the system links entities in the input, retrieves relevant triples or subgraphs, and exposes this knowledge to the model as context. The explanation consists mainly of the retrieved evidence itself. A user can inspect which entities are recognized, which relations are retrieved, and whether the generated response is supported by the retrieved evidence.

KAPING [9] exemplifies this strategy: it retrieves KG triples associated with entities in the input and linearizes them into the prompt, allowing the LLM to answer with access to explicit factual evidence. From an explainability perspective, the retrieved triples are visible factual ground of the answer: if the model produces an unsupported claim, the mismatch between the answer and the retrieved KG evidence can be inspected directly. Retrieve–Rewrite-Answer [185] follows a similar grounding logic, adding an intermediate rewriting stage. Rather than feeding raw graph evidence directly into the LLM, the method retrieves relevant knowledge, rewrites it into a form more suitable for generation, and then produces the final answer. This separation is useful for explainability because each stage can be inspected: what was retrieved, how it was rewritten, and how the final answer relates to the rewritten evidence. Rigel Facts [142] improves LLM answering by connecting generation to structured KG information, shifting part of the explanatory burden from the opaque model to the external knowledge source. The answer is no longer justified only by the model's internal confidence or fluency, but by explicit graph evidence that can be reviewed.

More recent GraphRAG approaches extend grounding by using the KG not only as a source of isolated triples, but as a structure for organizing retrieval [61, 127]. For instance, KG^{2}RAG [217] uses KG

Table 3 Overview of evidence-based explainability categories for LLMs: KG-based and RAG-based approaches.

Approach	Category	Main XAI Question	Typical Methods	Representative Work
KG Based	Knowledge Retrieval and Grounding	What evidence supports the prediction?	Retrieved triples, linked entities, KG neighborhoods, structured query results, provenance records	KAPING [9], Retrieve–Rewrite–Answer [185], Rigel Facts [142], StructGPT [75], KG^{2}RAG [217]
	Knowledge-guided Reasoning	How was the prediction derived?	Graph paths, traversal traces, symbolic queries, rule applications, graph-guided CoT steps	Think-on-Graph [156], DCRAG [197], RoG [103], MindMap [180]
	Knowledge-based Verification	Is the prediction consistent with external knowledge?	Supported, unsupported, contradicted labels, conflicting triples, constraint violations, corrective evidence	FactGraph [137], KEA [63], FLEEK [47], FactAlign [133], GraphEval [140]
RAG Based	Evidence Grounding	What external evidence supports the generated output?	Citation generation, claim–source attribution, fine-grained evidence selection, provenance tracing	ALCE [51], AIS [134], Attribute First then Generate [150], FRONT [69], MIRAGE [130]
	Reasoning over Retrieved Evidence	How was retrieved evidence used during inference?	Iterative retrieval, retrieval-conditioned reasoning	IRCoT [163], TRACE [46]
	Evidence Verification and Refinement	Should the retrieved evidence be accepted, rejected, or revised?	Evidence critique, relevance and support assessment, source reliability estimation	Self-RAG [6], CoV-RAG [64], RA-RAG [71]

structure to improve retrieval-augmented generation by expanding, connecting, or filtering evidence through graph relations. This also improves the explanatory value of RAG: instead of presenting a flat list of passages, the system can show how retrieved facts are connected through entities and relations, producing more structured and interpretable evidence than standard text-based RAG. StructGPT [75] is a relevant approach in this context, although not limited to KG support alone. This approach enables LLMs to interact with structured data sources through iterative querying. When the structured source is a KG or knowledge base, the executed queries and returned results can be inspected as grounding artifacts, demonstrating a transition from passive retrieval to structured interaction between KG and LLM.

Similar approaches extend grounding to more interactive or domain-specific contexts. IKA [26] combines language models with traditional graph networks for processing graph data and grounding LLM generation, while MedKit [155] grounds biomedical report generation in specialized medical knowledge sources. Dual-Reasoning [95] further structure in-context grounding from retrieved triples using cognitively inspired mechanisms such as dual-process theory, making the selection and use of knowledge more efficient and interpretable.

The main explainability contribution of knowledge retrieval and grounding is traceability: LLM outputs can be connected to retrieved facts, entities, and relations. However, grounding alone does not guarantee faithful reasoning: a model may ignore the retrieved triples, use them selectively, or generate claims beyond the available evidence. Thus, retrieval and grounding methods answer the question "what evidence was available?", but not always "how was that evidence used?"

Knowledge-Guided Reasoning The second role of KGs is to make the reasoning process itself more transparent. Here, the KG is not only a source of supporting facts, but a structure over which the LLM can reason. Explanations are produced as graph traversals, multi-hop paths, relation chains, or graph-guided CoT steps.

Think-on-Graph [156] is one of the most representative examples. Instead of relying only on free-form reasoning, the model iteratively explores relevant nodes and relations in a KG. The resulting reasoning process can be represented as a graph path or sequence of graph operations. The graph path is valuable for explainability because the intermediate steps are grounded in external symbolic structures rather than natural-language rationales generated by the model. RoG (Reasoning on Graphs) [103] similarly emphasizes faithful and interpretable reasoning over graph structures. The central idea of this approach is that multi-hop reasoning is grounded in explicit graph paths, e.g., if an answer depends on connecting entity (A) to entity (D), the explanation can show the path: $A \xrightarrow{r_1} B \xrightarrow{r_2} C \xrightarrow{r_3} D$. This type of explanation is especially useful for com-

plex question answering, where the final answer may be correct but difficult to trust unless the intermediate relational steps are visible. IRCoT [163] interleaves retrieval with CoT reasoning. When retrieval is structured or KG-assisted, each partial reasoning step can trigger the exploration of additional entities or relations, producing an iterative evidence-and-reasoning loop.

Other systems construct richer reasoning graphs. MindMap [180] combines implicit parametric knowledge with explicit KG structure, exposing a graph-like representation of how concepts, entities, and facts used to produce the final output are connected. DCRAG [197] similarly combines implicit knowledge with explicit knowledge from external KG triples by identifying the knowledge required for reasoning and retrieving relevant subgraphs to support the LLM's inference process.

This category also includes methods that rely on structured tool use and semantic parsing to iteratively explore KG during LLM reasoning, including systems that generate SPARQL, Cypher, or other graph queries during reasoning to produce supporting auditable traces [152, 153, 212]. Graph-ToolFormer [206] equips LLMs with graph tools, enabling them to call graph operations during reasoning and produce query or traversal logs. These logs are valuable explanatory artifacts: unlike unstructured CoT rationales, they correspond to operations executed on an external symbolic structure, thus providing a verifiable record of how the model accessed and manipulated structured knowledge to arrive at its answer.

Constrained decoding methods also constitute a form of knowledge-guided reasoning by incorporating symbolic constraints derived from external KGs into the LLM inference process. These methods integrate symbolic knowledge directly into token generation, restricting the model's output space to predictions that are consistent with KG facts or ontological axioms [104, 128].

In conclusion, graph-guided CoT frameworks leverage KG structure to constrain intermediate reasoning steps. Compared with unconstrained CoT prompting, graph-guided reasoning has an important explainability advantage: the reasoning trace can be easily checked against symbolic evidence. While unstructured rationales may be plausible but fabricated, a graph path, query log, or relation sequence can be fully and independently verified.

Therefore, the main contribution of knowledge-guided reasoning is transparency. These methods support LLM inference and answer to the question "how was the prediction derived?" by exposing intermediate symbolic structures. Their limitation is that graph reasoning can become complex or brittle, i.e., if entity linking fails, if the KG is incomplete, or if the model chooses an irrelevant path, the explanation may look structured while still being misleading. For this reason, reasoning traces should be evaluated not only for readability, but also for faithfulness and factual correctness.

Knowledge-Based Verification The third role of KGs is to verify generated outputs after or during generation. In this setting, the KG acts as an external reference against which the model's claims are checked. The explanation does not answer anymore the question "how did the model reason?", but rather "why should this answer be trusted or rejected?"

KG-based factuality verification methods frameworks are the most direct example. These systems typically extract atomic claims from generated text, link their entities and relations to a KG, and compare them with known triples or constraints. The resulting explanation verifies that a claim is supported, contradicted, or unsupported by a particular triple. Graph-level methods such as KEA [63] and FactGraph [137] compare generated content with KG subgraphs and assess whether the structure of the generated answer is compatible with known facts. Triple-level methods verify atomic claims against individual KG triples, often combining graph matching with entailment checks [133, 140]. FLEEK [47] leverages an LLM-as-a-judge pipeline for claim verification, generating targeted validation questions and retrieving mixed KG and Web evidence. The study of KG alignment with LLMs verifies whether retrieved evidence and generated claims are consistent with KG structure, rather than assuming that RAG pipelines are always reliable [162]. In particular, the literature highlights that ad-hoc tokenization and model-specific ordering of KG structures can improve relational alignment in LLMs, allowing retrieval pipelines to supply more precise and traceable evidence for inference and subsequent verification of generated claims.

Graph-based consistency checking extends the role of knowledge graphs from passive knowledge repositories to active sources of symbolic evidence for verifying and correcting LLM outputs. Generated claims can be compared against graph neighborhoods, relation constraints, ontology axioms, or known facts to assess their factual consistency. This capability is particularly valuable in domains such as biomedicine, law, and finance, where validity is governed by explicit symbolic relationships, e.g., a biomedical response can be verified against drug–disease or contraindication relations, with the KG explaining corrections by providing evidence such as "Drug A is contraindicated for Disease B."

Verification can also be coupled with runtime correction. When a claim is identified as contradicted or unsupported, the KG provides an explanation to support

the generation of a revised statement. In this setting, the KG functions not only as a fact checker but also as an explanatory tool: entities, relations, and graph paths provide an explicit justification for why the original claim was rejected and why it requires revision. Therefore, the main contribution of knowledge-based verification is diagnostic explainability: the explanation supports both human interpretability, by making the correction process transparent and traceable, and model-level reasoning, by providing structured evidence for refinement and evidence-aware prediction.

Summary of KG-Based Explainability Knowledge retrieval and grounding provide precise, fact-level evidence, but only for facts the KG contains and for entities that are correctly linked [9]. In knowledge-guided reasoning, graph paths and query logs can be checked hop by hop against the graph [103, 156], but missing edges or wrongly linked entities produce well-formed but misleading paths [43]. Knowledge-based verification confirms truthfulness of claims with symbolic verdicts (e.g., supported, contradicted, unsupported) [63], although absent triples must not be read as contradictions [43]. Finally, grounding and verification can support domains with a curated graph or ontology available, in which validity is defined by explicit relations, such as drug–disease interactions or regulatory constraints, while knowledge-guided reasoning suits multi-hop scenarios where each intermediate step must be auditable.

> **Key Takeaway.** Supporting a model with symbolic representations explains what evidence was available to it, but not necessarily how the information was used to produce the output.

4.1.2 RAG-Based Explanations

Whereas knowledge graph-based approaches ground LLM inference in structured symbolic knowledge, RAG-based approaches improve transparency by connecting generated outputs to external textual evidence. From an explainability perspective, retrieved evidence can play different roles: it can provide traceable support for generated claims, participate explicitly in intermediate reasoning, or be evaluated and refined during inference. Table 3 provides an overview of approaches for RAG-based explainability.

Evidence Grounding Rather than relying exclusively on knowledge encoded in model parameters, RAG systems retrieve external documents or passages and make them available during generation. Retrieval alone, however, is insufficient for explainability: the retrieved evidence must be explicitly connected to the claims it is intended to support. The resulting explanation therefore takes the form of an attribution between generated content and identifiable external sources, allowing users to trace the origin of information.

ALCE [51] provides a representative example of this approach by studying long-form generation with explicit citations to retrieved sources. Beyond answer correctness, ALCE evaluates whether cited sources entail the associated claims and whether claims requiring external support are sufficiently covered by citations. More generally, the Attributable to Identified Sources (AIS) framework [134] formalizes attribution as the requirement that externally verifiable claims be supported by identifiable sources, providing a conceptual foundation for treating source attribution as an explanatory mechanism. Subsequent approaches make the correspondence between claims and supporting evidence more fine-grained. *Attribute First, then Generate* [150] reverses the conventional generate-then-cite process by first selecting concise source segments and then generating text conditioned on those attributed segments. Similarly, FRONT [69] moves beyond coarse document-level citations towards fine-grained quotations that expose the specific portions of retrieved sources supporting individual claims.

Nevertheless, explicit citations do not necessarily provide a faithful account of how retrieved context influenced generation. A model may cite a relevant source without actually relying on it, or may use retrieved information that is not reflected in its citations. MIRAGE [130] addresses this distinction by applying model-internal attribution techniques to identify which retrieved documents contribute to individual answer tokens. Evidence-grounding approaches therefore improve source-level provenance and traceability, but do not necessarily reveal how retrieved evidence is combined and used during inference.

Beyond textual documents, this principle of semantic grounding is increasingly applied to multimodal recommender systems, such as e-commerce video feeds. By shifting from opaque collaborative filtering to multimodal retrieval that maps users and videos into a shared semantic space, recommendations can be grounded in explicit visual and semantic similarity rather than purely behavioral history, thereby providing a transparent basis for why a specific item was suggested [112, 113].

Reasoning over Retrieved Evidence Reasoning-aware RAG approaches interleave evidence acquisition with inference, producing trajectories in which intermediate reasoning identifies what information should be retrieved

next and newly retrieved evidence, in turn, informs subsequent steps.

IRCoT [163] is a representative example of this paradigm. The method addresses multi-step questions by interleaving Chain-of-Thought reasoning with retrieval rather than relying on a single initial retrieval step. Each intermediate reasoning step is used to guide the next retrieval operation, while the newly retrieved evidence provides additional context for subsequent reasoning. Because the generated reasoning steps directly influence which evidence is retrieved next, the interaction between reasoning and evidence acquisition becomes inspectable, although the resulting trace should not be interpreted as a faithful representation of the model's internal reasoning process.

This idea can be extended from iterative retrieval towards explicit evidence-based reasoning structures. TRACE [46], for instance, first transforms retrieved documents into a knowledge graph composed of entity–relation–entity triples. An autoregressive reasoning-chain constructor then iteratively ranks candidate triples and selects the next piece of evidence conditioned on the question and the previously selected triples, producing a sequence of logically connected supporting facts. The resulting chain can either be supplied directly to the reader or used to recover the corresponding source documents for answer generation.

However, while retrieval operations and their returned evidence are externally observable and can be independently verified, the free-text reasoning connecting them remains model-generated and inherits the distinction between plausibility and faithfulness discussed in Section 3. Consequently, reasoning-aware RAG improves process-level transparency by exposing how evidence is acquired and organized during inference, while the faithfulness of the natural-language reasoning connecting these operations remains an open question.

Evidence Verification and Refinement Evidence verification and refinement approaches evaluate whether retrieved evidence is appropriate for supporting a generated output, using the assessment to adapt subsequent retrieval or generation. These methods introduce explicit evaluation signals into the retrieval–generation loop, making evidence assessment an explanatory artifact and a mechanism for improving model behavior.

Self-RAG [6] exemplifies this approach by incorporating special reflection tokens through which the model determines when retrieval is necessary and evaluates retrieved passages and generated responses in terms of properties such as relevance and support. At inference time, these reflection tokens are generated alongside the response and are used to decide whether retrieval should be invoked, assess the relevance and support of retrieved passages, and score candidate generations, thereby allowing the model to adaptively control both evidence acquisition and answer generation. These signals expose intermediate judgments about the relationship between evidence and generation rather than treating all retrieved information as equally useful.

CoV-RAG [64] extends this principle through explicit verification of both retrieval and generation. Its *chain-of-verification* mechanism evaluates retrieved evidence and the resulting response, allowing inadequate retrieval or inconsistent generation to trigger rewriting and refinement. The resulting verification judgments provide a diagnostic signal for identifying problematic retrieval or generation steps, while simultaneously guiding their correction. In conversational and database exploration settings, this verification can also be externalized to the user through explicit intent scoping and paraphrasing, ensuring the model's planned retrieval and reasoning align with the user's actual goals before the final response is generated [41].

Verification also concerns the reliability of the evidence source itself, rather than only its semantic relevance. RA-RAG [71] incorporates source-reliability assessment and cross-source comparison into retrieval-augmented generation, addressing the possibility that highly relevant information may nevertheless originate from unreliable or conflicting sources. RA-RAG estimates source reliability by cross-checking answers across multiple sources, then selectively retrieves from the most reliable and relevant sources, aggregating their evidence by weighted majority voting, so that more trustworthy sources exert greater influence on the final response.

Together, these methods expose explicit signals about why retrieved evidence should be accepted, rejected, supplemented, or revised. RAG-based explainability thus progresses from source-level traceability, through evidence-mediated reasoning, to active verification and correction, illustrating how explanations can serve not only to make LLM outputs more inspectable but also to improve the reliability of the predictor itself.

Summary of RAG-Based Explainability Evidence grounding offers broader coverage than KGs without the need for schema design, allowing users to trace the origin of information [51], but a citation to retrieved passages alone does not establish that the passage was actually used by the model [130]. Reasoning over unstructured evidence can expose which evidence was acquired at each step [163]; however, unlike KG paths, links between passages cannot be checked symbolically. Verification and refinement assess relevance, support, and source reliability and can trigger

correction [6, 64, 71], but the lack of checks against explicit facts reduce the effectiveness of verdicts [45].

While KG-based methods suit scenarios where a curated KG justifies its construction and maintenance cost, RAG-based explainability supports large, unstructured, or frequently updated document collections, such as case-specific documents, recent literature, or news, that a KG cannot track without continuous re-extraction.

4.2 Retroactive Explainability

Whereas evidence-based methods make a predictor more transparent by grounding its inference in external structure, retroactive explainability methods intervene after the predictor has acted. They retrospectively generate an explanation of a prior outcome, such as a critique, diagnosis, or distilled strategy, and feed it back into the system to influence subsequent behavior. The explanation thus serves a prospective function, acting as an in-context signal, reward signal, or training signal for the next attempt.

Existing surveys of self-improving and self-correcting LLMs organize methods by training stage or by the presence of external feedback [77]. In the context of explainability, the relevant question is instead what carries the explanatory signal and where it re-enters the model. Retroactive methods are accordingly organized into four functional categories: *reflective diagnosis*, where a verbal account of a failure is appended to the prompt; *experience distillation*, where the lessons of past episodes are stored and retrieved across tasks; *critique-based evaluation*, where a dedicated model explains and scores outputs to supervise training; and *explanation-driven training*, where explanations are used directly as training data (Table 4). The categories are ordered by the permanence of the explanatory signal, from a transient note in the context window to a change in model weights. They are not mutually exclusive. Reflexion [146], for instance, combines a verbal diagnosis with an episodic memory, and Constitutional AI [10] uses model-written critiques both as revision targets and to supervise a preference model. The taxonomy therefore identifies the primary explanatory function of each method.

> **Key Takeaway.** Once an explanation of a past outcome is fed back into the model, it becomes an input that can influence the model and shape what it will do next.

4.2.1 Reflective Diagnosis

The first role of a retroactive explanation is to diagnose a failed attempt in natural language and make that diagnosis available to the next attempt. The explanation consists of the diagnosis itself: a statement of what went wrong and, ideally, why. Methods in this category differ in what produces the diagnosis and in how much of the cause it states.

Reflexion [146] exemplifies the self-generated variant. An agent acts, is evaluated, reflects verbally on the outcome, and stores the reflection in an episodic memory that is prepended to the next trial. The method reports state-of-the-art results on coding and embodied-agent tasks, and an ablation attributes more of the gain to the verbal reflection than to the episodic memory alone. Self-Refine [105] applies the same generate-critique-revise pattern to single-turn generation, with one model serving as generator, critic, and reviser. A second group replaces self-critique with a trained critic. Self-Correct [179] trains a separate corrector for single-turn generation; Retroformer [194] trains a small retrospective model for multi-step agents by policy gradient on the environment's reward while keeping the actor frozen; and RetroAct [48] trains planner and reflector jointly through imitation learning followed by off-policy reinforcement learning. CRITIC [56] routes the critique step through external tools such as search engines and code interpreters, so that the diagnosis rests on tool output rather than on the model's own judgment. On the content of the diagnosis, Renze and Guven [135] compare a bare retry signal against progressively richer reflections, finding that an explicit account of why an attempt failed is more beneficial than a signal that only indicates failure. This finding suggests that the explanatory content, rather than the additional attempt alone, drives the gain.

The main explainability contribution of reflective diagnosis is error localization: a failure is named, and the next attempt is conditioned on that name. However, the accuracy of the diagnosis is not guaranteed. Models are not reliably better at judging their own candidate outputs than at producing them [74], and iterated self-refinement amplifies self-bias, raising the model's own assessment of its revisions without a corresponding gain in quality [187]. Reflexion's authors [146] note that part of its code-generation improvement rests on self-generated unit tests, which may encode the same mistake as the code they are meant to check, and that the agent fails to produce useful reflections on tasks requiring wide exploration. Self-Refine's [105] gains are much larger on stylistic tasks than on reasoning-heavy ones. These observations agree with the broader find-

Table 4 Overview of retroactive explainability categories for LLMs.

Category	Main XAI Question	Typical Methods	Representative Work
Reflective Diagnosis	Why did the previous attempt fail?	Natural-language self-critiques, diagnoses from a trained retrospective model, tool-grounded critiques	Reflexion [146], Self-Refine [105], Self-Correct [179], Retroformer [194], RetroAct [48], CRITIC [56]
Experience Distillation	What in past episodes is worth reusing?	Abstracted trajectories, induced workflows, code skills, strategy summaries, cross-task insights	Synapse [216], AWM [174], Voyager [166], Dynamic Cheatsheet [157], ReasoningBank [123], ExpeL [210], ReMe [20], ACE [208]
Critique-Based Evaluation	How good is this output, and why?	Step-level scores, written critiques paired with scores, self-judgments used as rewards	Shepherd [171], Math-Shepherd [170], CriticGPT [108], Critic-RM [199], Self-Rewarding LMs [201]
Explanation-Driven Training	What should the model learn from this outcome?	Rationales and rationalizations, mistake–correction pairs, principle-guided revisions, textual gradients	STaR [204], V-STaR [66], ReST^{EM} [148], LEMA [5], Constitutional AI [10], RLAIF [89], SCoRe [85], TextGrad [202], ProTeGi [129]

ing that successful self-correction from prompted-LLM feedback alone has not been demonstrated outside tasks especially suited to it, whereas reliable external feedback does help [77]. A trained critic avoids self-bias but not the discrimination limit; a tool-grounded critic avoids both, at the price of trusting the tools. In all cases the diagnosis is a form of chain of thought about a past output, and it inherits the gap between plausibility and faithfulness discussed in the previous section [28, 164]. Thus, reflective diagnosis answers the question "what went wrong?" in a form the model can act on, but not the question "is that account true?".

4.2.2 Experience Distillation

The second role of a retroactive explanation is to distill past episodes into a reusable form that is retrieved when a similar task recurs. Here the explanatory act is the distillation itself: deciding what to keep from an episode is a claim about what mattered in it. The explanation consists of the stored artifact.

Early instances are Generative Agents [126], which synthesize observations into higher-level reflections, and CLIN [107], which accumulates causal abstractions across trials. The category is best organized by the level of abstraction at which the artifact is kept. Synapse [216] sits at the least-abstracted end, retrieving whole past trajectories as few-shot exemplars after a state-abstraction step that trims raw observations to what is task-relevant; the explanation it offers is limited to "here is a similar situation ", but since the retrieved trajectory is stored almost verbatim, it provides an example to imitate but no indication of which of its steps were responsible for success. Agent Workflow Memory (AWM) [174] addresses this by abstracting over several successful trajectories: it extracts the steps they have in common and replaces episode-specific values, such as a product name or a URL, with variables, yielding a reusable procedure that applies to any task of the same type. Voyager [166] does the equivalent in an embodied setting with a library of executable code skills. Dynamic Cheatsheet [157] accumulates strategies and code snippets across queries at inference time without labels. At the most abstract end, ReasoningBank [123] stores a titled strategy summary rather than the trajectory, Agent KB [159] organizes such summaries hierarchically for cross-domain transfer, and Metacognitive Reuse [37] distills recurring reasoning patterns into named behaviors. Off this axis, Contextual Experience Replay [98], A-MEM [188], and Mem0 [30] vary the organization of the store rather than its abstraction, through a single dynamic buffer, agent-generated links between notes, and consolidation of conversational history, respectively.

Two further distinctions cut across this axis. The first is whether failed episodes are distilled at all. Most systems store successes; ExpeL [210] contrasts successful and failed trajectory pairs, ReasoningBank retains failures for the cautionary pattern they contain, and ReMe [20] extracts success patterns, failure triggers, and comparative insights as separate experience types. The second distinction is whether the store is maintained rather than only appended to. Agentic Context Engineering (ACE) [208] tracks how often each entry proved helpful or harmful and separates generation, reflection, and curation to avoid collapsing the store into a short summary; ReMe prunes entries by estimated utility; Dynamic Cheatsheet [157] curates and rewrites its memory rather than extending it. In these systems, the curation

step is itself a claim about which past lessons still hold.

The main explainability contribution of experience distillation is transferable abstraction: the system commits to a statement of which aspects of an episode generalize. However, moving from trajectory to procedure to strategy trades fidelity to what happened for transfer to dissimilar tasks, the same trade-off noted for human-facing explanation earlier in this survey. All of these systems report downstream gains, but none checks directly whether the distilled content is the reason an episode succeeded or failed rather than something correlated with it; ReasoningBank's self-judged success labels are a step towards such a check, not the check itself. The scarcity of memory systems that mine failures is also notable given that improvement from past outcomes is the stated purpose of the category. Experience distillation therefore answers the question "what is worth reusing?" but not "why did it work?".

4.2.3 Critique-Based Evaluation

The third role of a retroactive explanation is to evaluate an output through a dedicated critique or reward model whose judgment, and ideally its justification, supervises further training. The explanation consists of the critique attached to a score. Methods in this category differ in the granularity of the critique and, more consequentially, in the source of its supervision.

On granularity, process reward models score each step of a reasoning chain. Lightman et al. [92] show that step-level scoring produces a markedly more reliable verifier than scoring only the final answer, using human labels at every step. Response-level critics, by contrast, assign a single judgment to the complete response, without localizing the error to a particular step; their explanation therefore concerns the output as a whole rather than the point at which the reasoning went wrong. Shepherd [171] generates a written critique of a response, and CriticGPT [108] and Critic-RM [199] pair such a critique with a score, so that the reward model has to justify its number rather than only emit it. On supervision, the category traces a progressive retreat from human grounding. The earliest critic models [141, 171] and the original process reward model are trained on human-written critiques or human step labels. Math-Shepherd [170] replaces step labels with estimates derived from sampled rollouts. CriticGPT is trained with RLHF on bugs that human contractors deliberately planted in model-written code; its inference-time search trades comprehensiveness against hallucinated bugs, whose rate the authors describe as reduced but still high, and human–model teams hallucinate less than the critic alone. Critic-RM keeps only an indirect human signal, filtering model-written critiques for agreement with existing preference data. Self-Rewarding Language Models [201] remove the human entirely: one model acts as both generator and judge across repeated rounds of DPO training, and improves over its seed model on AlpacaEval 2.0. Meta-Rewarding [184] subsequently finds that this signal saturates quickly unless the judge is itself trained, and adds a meta-judge for that purpose.

The main explainability contribution of critique-based evaluation is justified supervision: the signal that shapes training carries a stated reason, which can in principle be audited. However, the value of that reason depends on what grounds it. Once no external check remains, there is no way to determine whether a critique identifies a real flaw or merely reads as if it does, and self-correction in reasoning is known to be unreliable precisely when such grounding is absent [68]. The granularity of the critique does not resolve this; a step-level critic without grounding is ungrounded at a finer resolution. For this reason, the trend towards removing the human from the loop is also a trend towards critiques whose faithfulness cannot be established.

4.2.4 Explanation-Driven Training

The fourth role of a retroactive explanation is to serve directly as training data, so that the loop closes with a gradient update. The explanation consists of a rationale, a revision, or a critique that becomes the training target. Methods in this category differ in what is distilled and in how the update is computed.

Four kinds of artifact are distilled. The first is rationales for correct answers. STaR [204] keeps self-generated rationales that reach the correct answer and, for incorrect ones, regenerates a rationale with the answer supplied as a hint, a step the authors term rationalization; the hint is then dropped, and the model is fine-tuned as if it had produced the rationale unaided, and the authors themselves note that nothing ensures such rationales reflect the model's internal processing. ReST^{EM} [148] drops the rationale step altogether and fine-tunes on self-generated samples filtered for correctness over several rounds; it still outperforms training on human data alone. The second kind is negative examples: V-STaR [66] adds incorrect self-generated solutions to train a verifier. The third is explained failures. LEMA [5] fine-tunes on mistakes, explanations, and correction triples produced by a stronger model, and Critiques Fine-Tuning [173] trains on critiques of noisy responses rather than on clean responses. Because generating and training on all possible failures is inefficient, selecting the most informative mistakes to correct naturally aligns with active learning and reinforced sampling strategies,

ensuring the policy focuses on the highest-value updates rather than redundant errors [181, 182].

The fourth is principle-guided revisions. Constitutional AI [10] proceeds in two stages. In the first, a model critiques its own harmful response against a written principle, rewrites it, and is fine-tuned on the rewrite. In the second, the model compares pairs of responses against the same principles; these AI-generated preference labels are used to train a preference model, and that preference model then serves as the reward signal for reinforcement learning on the policy. RLAIF [89] tests that second idea outside alignment and finds that a reward model trained on AI preferences matches one trained on human preferences on summarization and dialogue, even when the labeler is the same size as the policy. On the update itself, SCoRe [85] finds that supervised fine-tuning on self-generated correction traces, the natural extension of the STaR and ReST^{EM} recipe to correction data, often fails to teach general self-correction, because training-time errors do not match test-time errors and the model learns a narrow correction habit; its remedy is multi-turn reinforcement learning on the same traces. At the limit of this category, the explanation becomes the optimizer. TextGrad [202] and ProTeGi [129] treat a natural-language critique as a textual gradient that updates a prompt, a program, or another free-text variable.

The main explainability contribution of explanation-driven training is persistence: the explanation changes the model rather than a single output. However, this is also the category where the explanatory content is hardest to isolate. ReST^{EM}'s result raises the possibility that the correctness filter, not the explanation, produces the gain, and neither LEMA nor critique fine-tuning disentangles the explanation from the corrected answer it accompanies. The SCoRe result shows that how the update is computed matters as much as what is distilled. Scale matters as well: Constitutional AI reports harmlessness improving monotonically with each critique-revision round at its original scale, whereas a replication on an 8B model [209] found a 9-point loss in helpfulness for the harmlessness gained and degenerate outputs that the authors describe as model collapse, echoing the scale sensitivity observed for Self-Refine and Self-Rewarding. Explanation-driven training thus answers the question "what should the model learn?" with a permanent update, while leaving open whether the explanation, rather than the filtering or the reward, is what was learned.

5 Explainability in Domain-Specific Applications

The requirements of explainability depend heavily on the domain in which an AI system is deployed. Different techniques expose different types of information, and their usefulness therefore depends on the forms of evidence, reasoning, verification, and intervention required by the application. This becomes particularly important in high-stakes settings. In healthcare, evidence-based explanations can expose the clinical evidence supporting a recommendation; in law, explanations may need to connect claims to applicable statutes or precedents; and in finance, provenance is often complemented by the need to reconstruct numerical calculations. Domain-specific explainability therefore concerns not only whether an explanation is interpretable, but whether the chosen technique exposes the information required for the evaluation and oversight by domain-specific experts.

> **Key Takeaway.** Different high-stakes domains require different kinds of evidence from an explanation, and no single method can serve every specific domain equally well.

5.1 Healthcare and Biomedical Applications

Healthcare provides a particularly relevant setting for domain-specific explainability because different explanation techniques address different clinical requirements. Clinical decision-making also relies on heterogeneous and often longitudinal evidence, including structured measurements, clinical text, medical images, and observations collected at different points in time. Explanations may therefore support understanding of the model's internal behavior, inspection of its reasoning, identification of supporting clinical evidence, or verification of its recommendations, while also revealing how information across modalities and over time contributes to a decision. These functions are closely related to clinical trust: explainability can either increase or decrease clinicians' trust depending on the quality of the explanation, and increased trust is not necessarily desirable when the underlying recommendation is incorrect [139].

Evidence-based approaches supported by RAG and knowledge graphs represent a prominent direction in healthcare [97] . Their relevance follows directly from the evidence-based nature of clinical decision-making: provenance as a form of explainability is a common choice. CLEAR [99], for example, retrieves clinically relevant EHR passages through medical entities, while RAG^2 [151] uses rationale-guided retrieval and filtering to improve the relevance of biomedical evidence.

Knowledge graphs extend this principle by making relationships between pieces of evidence explicit. For instance, MedRAG [214] combines retrieved EHRs with a diagnostic knowledge graph that encodes differences between diseases with similar manifestations. MedGraphRAG [183] introduces a graph-based retrieval mechanism specifically designed for medical knowledge. It constructs a triple-linked graph connecting user-provided documents, credible medical sources, and controlled medical vocabularies, and introduces U-Retrieval, which combines top-down precise retrieval with bottom-up response refinement.

Reasoning-based explanations are also attractive in clinical applications because they expose intermediate steps connecting patient information to clinical recommendations. However, their usefulness remains constrained by the distinction between plausibility and faithfulness: a clinically coherent reasoning trace does not necessarily reflect the process that produced the prediction. Recent evaluations further show that medical models can generate convincing reasoning traces even when those explanations rely on inaccurate or insufficient clinical evidence [57]. Mechanistic interpretability remains comparatively recent in healthcare. Tan et al. [158] analyze medically fine-tuned LLMs using transcoders and attribution graphs, identifying internal pathways associated with pattern matching and multi-hop diagnostic reasoning.

Similarly, retroactive explainability has only recently begun to appear in biomedical applications. Self-BioRAG [73] incorporates reflective signals that allow the model to evaluate generated responses and determine when additional external evidence should be retrieved. More recent work extends this idea through iterative self-critique and correction, as in MCQG-SRefine [196], Med-REFL [191], and MedReflect [70]. These methods move towards explainability for the predictor, since signals obtained from evaluation or reflection influence subsequent reasoning and correction.

Multimodal and longitudinal clinical data introduce an additional explainability challenge: explanations must account not only for evidence within individual modalities, but also for how clinically relevant information is connected across modalities and evolves over time. Tracking these changes often requires robust similarity matching and correlation discovery over uncertain temporal data before an LLM can reliably explain disease progression [34, 111]. Recent medical vision–language approaches illustrate this through explicit grounding of clinical descriptions in the underlying images. TempA-VLP [190], for example, introduces dynamic phrase grounding, linking descriptions of disease progression to corresponding regions across consecutive chest X-rays. Diff-RRG [203] similarly models disease-specific patch differences between current and prior radiographs and uses these localized changes to guide LLM-based report generation, allowing generated descriptions of disease progression to be traced back to relevant image regions.

Overall, healthcare shows a strong emphasis on evidence-based explainability, reflecting the importance of provenance and external verification in clinical decision-making. Reasoning-based explanations provide a more accessible interface but remain limited by the open faithfulness challenge, while multimodal and longitudinal settings introduce the additional need to connect evidence across heterogeneous sources and observations over time. Mechanistic and retroactive explainability approaches represent more recent directions for auditing and improving medical LLM behavior.

5.2 Legal Applications

Legal applications provide a particularly relevant setting for domain-specific explainability because the validity of a model's output depends on the relationship between factual evidence, authoritative legal sources, and the resulting conclusion. Recent reviews of legal LLMs emphasize the importance of incorporating explicit legal knowledge—including statutes, cases, and judicial interpretations—to improve reliability and make the legal basis of generated outputs more inspectable and verifiable [79, 96]. Explanations in this domain must therefore support not only inspection of a generated rationale, but also verification of which legal authorities support individual claims and whether they have been applied appropriately.

Evidence-based approaches supported by document retrieval and structured legal knowledge represent a prominent direction in this literature. CitaLaw [207], for example, evaluates whether generated legal claims are actually supported by their cited authorities, highlighting that citation presence alone does not guarantee evidential support. LexRAG [91] similarly evaluates retrieval and grounded generation in multi-turn legal consultation, linking generated responses to relevant legal provisions. More structured approaches go beyond semantic retrieval by explicitly modeling fact-rule relationships; LSIM [192], for instance, constructs logical fact-rule structures to guide the retrieval and use of relevant legal knowledge.

Reasoning-based explainability is also actively explored, reflecting the inherently structured nature of legal argumentation. Chain of Logic [144] decomposes compositional legal rules into intermediate reasoning steps inspired by the IRAC framework, while Mishra et al. [114] evaluate legal reasoning step-by-step and

introduce measures of soundness and correctness to identify where reasoning failures occur. Yuan et al. [200] similarly use multi-agent decomposition to reason over complex legal theories. Importantly, however, recent reviews caution that generated reasoning traces should not automatically be interpreted as faithful explanations, since plausible chains of thought may rationalize incorrect conclusions [79].

Retroactive and intervention-oriented approaches have emerged more recently. Argumentative LLMs [93] represent legal decisions through structured supporting and attacking arguments, providing explanations that can be inspected and contested. LawThinker [189] introduces an Explore–Verify–Memorize process in which retrieved legal information is repeatedly checked for accuracy, fact–law relevance, and procedural compliance before reasoning continues. More recently, Parthenon Law [53] uses failures from previous legal-agent trajectories to update reusable knowledge, tools, and procedural skills, moving from explanation and verification towards improvement of subsequent system behavior.

Overall, the legal literature shows a strong emphasis on knowledge grounding and structured reasoning, reflecting the importance of authoritative sources and explicit fact–rule relationships in legal decision-making. Retroactive explainability approaches are beginning to extend these mechanisms towards verification and correction, whereas mechanistic interpretability of legal LLM reasoning remains comparatively unexplored.

5.3 Financial Applications

Financial applications provide another high-stakes setting in which explainability requirements are strongly shaped by the domain. Financial models operate over heterogeneous evidence, including annual reports, regulatory filings, financial statements, tables, and charts, while many tasks require explicit numerical calculations and comparisons. Recent systematic reviews of XAI in finance emphasize transparency, traceability, and verifiability as central requirements, particularly in settings where model outputs may inform consequential financial decisions [21, 175]. Explanations in this domain may therefore need to expose both the evidence supporting a financial claim and the intermediate quantities or operations used to derive it.

Evidence-based approaches supported by RAG represent an important direction in financial LLM applications. FinDVer [215], for example, evaluates explainable claim verification over long and hybrid-content financial documents, requiring models to identify evidence supporting or contradicting financial claims. FinRAGBench-V [213] extends this requirement to multimodal financial documents by incorporating visual citations, allowing generated claims to be traced to relevant tables, figures, and other visual evidence. More structured grounding has also been explored through financial knowledge graphs; Chen et al. [29] combine retrieval with a financial knowledge graph to support market analysis and improve the consistency of generated reports.

Financial reasoning benchmarks further highlight the importance of exposing intermediate computational structure. BizBench [84] separates financial problem solving into document comprehension, extraction of intermediate quantities, and application of financial concepts and formulas, while FinanceReasoning [160] targets more complex numerical problems requiring the composition of multiple financial formulas. Although these works primarily evaluate reasoning rather than explainability itself, they illustrate why intermediate quantities and operations can provide a more verifiable account of model behavior than linguistic rationales.

Retroactive approaches have emerged more recently. FinCon [198] incorporates episodic self-critique into financial decision-making, using reflections on previous investment decisions to update beliefs that influence subsequent behavior. CLER [36] similarly uses retrieved prior reasoning errors and step-wise reflection to improve multimodal financial reasoning, extending explanation-oriented mechanisms towards explicit error identification and correction.

Mechanistic interpretability remains comparatively less explored in financial LLMs. Preliminary work by Golgoon et al. [55] applies direct logit attribution and activation patching to identify internal components involved in financial-services compliance tasks, including potential Fair Lending violations. Such approaches demonstrate the potential of mechanistic analysis for model auditing and regulatory oversight, but remain substantially less established than evidence-grounded and reasoning-oriented methods.

Overall, financial LLM explainability is characterized by a strong emphasis on evidence grounding and verifiable reasoning. Retrieval-based approaches provide provenance over complex textual and visual financial documents, while reasoning-based methods expose intermediate quantities and calculations that can be independently checked. Retroactive explainability methods are emerging, moving the field towards error correction and improving subsequent decisions. Mechanistic interpretability remains a promising but comparatively niche direction.

Retrieval-based approaches provide provenance over complex textual and visual financial documents, while reasoning-oriented methods expose intermediate quantities and calculations that can be independently checked.

Table 5 Qualitative comparison of the relative presence and maturity of explainability approaches in the reviewed literature across high-stakes domains. *Strong*: well represented and a prominent direction within the domain; *Moderate*: established but less dominant body of work; *Emerging*: relatively recent direction represented by a limited but growing number of studies; *Absent*: little or no domain-specific work identified.

Domain	Mechanistic Interpretability	Reasoning-Based Explainability	Evidence-Based Explainability	Retroactive Explainability
Healthcare and Biomedical	Emerging	Moderate	Strong	Emerging
Legal	Absent	Moderate	Strong	Emerging
Financial	Emerging	Strong	Strong	Emerging

Together, these methods lay the groundwork for fully reliable conversational data analytics, where users can safely explore domain-specific data through iterative, verifiable interactions [4].

5.4 Cross-Domain Comparison

The comparison across healthcare, legal, and financial applications shows that the relevance and maturity of different explainability approaches vary substantially by domain. Evidence-based methods are prominent across all three settings because they provide external evidence that can be inspected and verified. Reasoning-based explanations are particularly important in legal and financial applications, where conclusions often depend on explicit rule-based or numerical reasoning, while their role in healthcare is more constrained by concerns about faithfulness and reliability. Retroactive explainability approaches are emerging across all three domains, whereas mechanistic interpretability remains comparatively less explored, with recent applications in healthcare and finance but very limited work in legal reasoning. Table 5 provides a qualitative characterization of these trends based on the literature reviewed in the previous sections.

6 Current Challenges and Research Directions

Each family of explainability methods produces an artifact that serves as an explanation: a circuit, a reasoning trace, a graph path, a citation, or a self-diagnosis. Each family also faces the same question about that artifact: whether it describes what the model did or merely accompanies it. This section discusses current challenges and research directions of explainability methods, including cross-cutting challenges shared among all families, challenges specific to each family, and research directions to improve explainability for LLMs.

> **Key Takeaway.** Whether the artifact is a circuit, a reasoning trace, external knowledge, or a self-diagnosis, the central question remains whether the explanation describes what caused the model's behavior.

6.1 Cross-Cutting Challenges

Causal Validation The central open problem of the field recurs across all families. In mechanistic interpretability, probes, attention patterns, and attribution scores show that information is encoded without showing that the model relies on it; decodable information does not imply causal use, and only interventions establish a causal role [145]. In reasoning-based interpretability, a chain of thought is optimized for plausibility under the next-token distribution rather than for fidelity to the internal computation, and the answer may be determined before the trace is generated [28, 164]. In evidence-based explainability, retrieval exposes which triples or passages were available but not whether they determined the output, since a model may ignore the evidence, use it selectively, or generate claims beyond it. In retroactive explainability, a self-generated diagnosis is fed back to the model regardless of whether it identifies the actual cause of failure. The artifact differs across families; the gap between correlation and causation does not. A validation technique developed for one family, such as activation patching or counterfactual evidence editing, is therefore applicable in principle to the others, but such transfer has not yet been undertaken systematically.

Evaluation No family possesses an accepted benchmark for the property it claims to provide. Mechanistic explanations lack ground truth for the contents of a circuit, so competing decompositions cannot be ranked [116, 205]. Chain-of-thought faithfulness is assessed through perturbation studies and behavioral probes, with no consensus on what constitutes a sufficient test [72, 149]. Evidence-based methods are evaluated on task accuracy, and the

faithfulness of the exposed evidence is rarely measured. Retroactive methods report downstream gains that do not separate the contribution of the explanation from that of the additional attempt, the correctness filter, or the reward signal. Three dimensions of explanation quality recur throughout the literature without being operationalized jointly: faithfulness to the computation, understandability to the intended reader, and granularity, that is, the level of abstraction at which the explanation is pitched. Results are moreover sensitive to decoding choices, so that reporting sampling temperature, the number of samples used for self-consistency, and any filtering applied to rationales is a precondition for comparison across studies.

Scale Findings obtained at one model scale do not transfer reliably to another. Most mechanistic studies concern small models and narrow behaviors; manual localization and exhaustive causal search do not scale to production models, which forces reliance on approximations such as attribution patching [132]. Retroactive methods exhibit the opposite sensitivity: Self-Refine, Self-Rewarding, and Constitutional AI behave differently on smaller models, in the last case to the point of model collapse [209]. Reasoning-oriented models trained with reinforcement learning introduce a further shift, since their verbalized reasoning diverges from their computation despite explicit training to reason step by step [28].

Audience Section 4 distinguishes explanations read by a human from explanations consumed by the system. In practice a single artifact is often required to serve both, and the two requirements conflict. Graph paths, traversal logs, and SPARQL queries are verifiable but presuppose familiarity with a schema and query language that clinicians and lawyers do not possess; translating them into natural language reintroduces the unverified rationale that the symbolic form was intended to replace. Conversely, a critique produced for a reward model or a strategy distilled for retrieval is never inspected by a person, so its faithfulness is neither checked nor checkable. As human supervision is removed from the loop, in the progression from human labels to self-rewarding, the artifact loses its only external validator.

6.2 Family-Specific Challenges

Mechanistic Interpretability Beyond scale and evaluation, the principal open question concerns the unit of analysis. Interpretable concepts frequently correspond to directions or subspaces in activation space rather than to individual neurons, and information is distributed across neurons, layers, and components, so that analyses framed around single units miss the relevant structure [49,145]. Learned decompositions do not resolve the question: sparse autoencoders exhibit feature absorption, and enforcing sparsity may discard mechanisms that are genuinely distributed. Greater automation of localization and hypothesis generation is required for the field to progress from individual case studies to systematic coverage [14].

Reasoning-Based Explainability Three challenges are specific to reasoning-based explanations. First, no method establishes a direct correspondence between the sequential structure of a trace and the distributed representations that produce the answer; this challenge connects this family of methods directly to mechanistic interpretability. Second, the interpretability desiderata compete: interventions that improve utility or reduce unfaithfulness do not consistently improve robustness, and no method optimizes the three dimensions jointly [177], leaving open whether these objectives are compatible. Third, since chain of thought is increasingly used to monitor agentic systems, the divergence between verbalized and actual computation constitutes an oversight problem in addition to the interpretability one [28]. Structured alternatives, such as chain-of-knowledge prompting [168] or program-of-thought [27], provide more auditable reasoning at some cost in expressiveness.

Evidence-Based Explainability For knowledge graph-based methods, structural form may offer false assurance: when entity linking fails, the graph is incomplete, or the model follows an irrelevant path, the explanation remains well-formed while being misleading. For instance, a word resolved to the wrong entity produces a confident verdict against the wrong triples, and the error propagates through retrieval, reasoning, and verification alike. Mitigating this false assurance requires highly robust entity matching, often necessitating specialized graph neural network architectures to accurately align concepts before the LLM reasons over them [81–83]. Additionally, incompleteness must be handled explicitly: missing triple should not automatically be interpreted as a false claim [43], and verification methods should report coverage limitations and distinguish unsupported claims from contradicted ones. For RAG-based methods, explanations are limited by the unstructured nature of text-based retrieval. The primary explanatory artifact is the set of retrieved passages, whereas KG-based methods can expose typed entities, relation paths, constraint violations, and provenance records. These structured artifacts are easier to audit and, in many cases, can be

verified formally; retrieved text offers no such counterpart, and verification typically falls back on entailment models and LLM-as-a-judge [45], which reintroduces the opacity that explainability was meant to remove. Faithfulness is a further concern: showing that a passage was retrieved does not establish that the model used it, and models frequently generate claims unsupported by their retrieved context [121]. Finally, the evidence set does not reveal how conflicts among redundant, outdated, or contradictory passages are resolved, and the retriever is itself an unexplained component.

Retroactive Explainability The defining operation of this family, feeding an explanation back into the system, gives rise to challenges the other families do not face. The first is attribution of the improvement. ReST^{EM} improves without any rationale, LEMA and critique fine-tuning do not separate the explanation from the corrected answer that accompanies it, and Reflexion's ablation is the only one that isolates the verbal signal from the additional attempt [5, 146, 148, 173]; for most of the family, whether the explanation or the surrounding loop produces the gain is unresolved. The second is the absence of an external reader. Self-generated diagnoses are subject to self-bias and to the discrimination limit [74, 187], distilled strategies are stored without any check that they identify the cause of success, and the progression from human step labels to self-rewarding removes the last independent check on the content of a critique; the diagnosis inherits the plausibility-faithfulness gap without anyone positioned to detect it. The third is the mismatch between what is learned and what is required at test time: supervised fine-tuning on self-generated corrections teaches a narrow correction habit rather than a general capacity, because training-time errors differ from test-time errors [85]. The fourth is that failures, the outcomes most in need of explanation, are the least often retained; most memory systems store successes, and the few that mine failures do so without validating the lesson extracted. Finally, this family is the most scale-sensitive of the four, and the conditions under which a self-improvement loop degrades rather than improves a model have not been characterized.

6.3 Research Directions

Interventional Validation The methods that establish causal role in mechanistic interpretability, namely activation and path patching, causal scrubbing, and counterfactual editing, are the natural instruments for the faithfulness question in the other three families. Patching the representations that a chain of thought claims to use would test whether the trace guides the answer or merely accompanies it. Editing retrieved evidence and observing whether the answer changes would establish whether grounding is causal rather than decorative. Applying the same edits to a self-diagnosis before it re-enters the prompt would test whether a retroactive explanation identifies the cause of a failure. Each of these is a direct transfer of an existing method to a new artifact, and none has been carried out systematically.

Hybrid Methods Knowledge graphs can ground the content of a reflection or a distilled strategy, replacing self-judged success labels with checkable claims; early work combining chain of thought with structured triples takes this direction [168], and comparative studies of prompting strategies identify it as promising [177]. Attribution methods can constrain what a textual explanation says: Faithfulness Serum [2] computes token-level relevance for the model's answer with layer-wise relevance propagation and injects it into the attention scores while the model generates its post-hoc explanation, which substantially increases how often the explanation acknowledges the input that actually changed the answer. The same principle could be applied to the diagnoses that retroactive methods consume, so that only those with a demonstrable causal basis are fed back. Retrieval can supply the external reader that self-critique lacks, as CRITIC does through tools [56]. The taxonomy of Section 2 maps what each family exposes; the hybrids of greatest value are those that pair an exposing method with a validating one.

Faithfulness-Oriented Training Reinforcement learning currently trains models to reason at greater length, not to report their reasoning accurately, and the divergence between the two is documented [28]. Two lines of work now treat faithfulness as a quantity to be optimized rather than only tested. At inference time, FaithLM [32] defines the faithfulness of an explanation as the shift in the model's prediction when a hint contradicting the explanation is added to the input, and iteratively rewrites both the explanation and the prompt that elicits it to maximize this score. At training time, Cheah et al. [24] convert an existing faithfulness metric into a per-sample reward and use reinforcement learning to update the model's parameters so that its self-explanations disclose the factors that influenced its decisions. Question decomposition further shows that faithfulness responds to how a task is posed [131]. These results remain limited to multiple-choice or classification settings with synthetic interventions, such as inserted hints or random words, and it is open whether they extend to open-ended generation and to long reasoning traces. The same principle applies to retroactive methods, where the reward could

depend on whether a diagnosis survives an intervention rather than on whether the subsequent attempt succeeds.

Shared Evaluation Protocols The three dimensions that recur across the survey, faithfulness, understandability, and granularity, should be measured on the same artifacts with the same tests regardless of the family that produced them. Such a protocol requires a faithfulness test that is interventional, an understandability test that names its intended reader, and an explicit statement of the level at which the explanation is pitched, together with the decoding and filtering choices under which it was obtained. Domain-specific settings shows that clinicians, lawyers, and analysts require different evidence, and a protocol that fixes the reader would make the qualitative comparison quantitative.

Conceptual Frameworks The dichotomy between transparent models and post-hoc explanations was inherited from earlier machine-learning paradigms and fits large auto-regressive models poorly, since in-context learning and emergent capabilities blur the boundary between the model and its inputs [7]. The framing adopted in this survey, from interpretation to intervention, is one alternative: an explanation is validated by what it permits one to do to the model, whether patching a circuit, correcting a claim against a graph, or conditioning the next attempt on a diagnosis. Making this criterion precise, and specifying the conditions under which an intervention counts as validation, remains conceptual work that the field has yet to undertake.

7 Conclusions

This survey has examined explainability for LLMs through a taxonomy that distinguishes *explainability of the prediction*, which accounts for a specific output, from *explainability for the predictor*, which characterizes and improves the overall model. This distinction reflects a shift in the role of explanations: from post-hoc artifacts for a human reader to signals that diagnose failures, guide updates, and ground outputs, useful both for humans and the system.

Within *explainability of the prediction*, mechanistic interpretability derives explanations from the computation itself and has progressed from the inspection of individual components to learned decompositions and circuit-level accounts, while its central question, whether a located component is causally responsible for a behavior, is answered only by intervention. Reasoning-based explainability derives explanations from generated text and offers a readable interface, but the literature reviewed here supports treating chain-of-thought traces as interpretable reasoning artifacts rather than as faithful accounts of the model's decision process. Within *explainability for the predictor*, evidence-based methods obtain transparency by grounding inference in knowledge graphs or retrieved documents, exposing what evidence was available, how it was traversed, and whether a claim is consistent with it. Retroactive methods feed explanations of past outcomes back into the system, through the prompt, a memory, a reward model, or a gradient update.

Analysis of three high-stakes domains (i.e., healthcare, law, finance) highlighted that RAG and KG-based are the most established methods, reasoning-based explanation is valued where conclusions rest on explicit rules or calculations, retroactive explainability is emerging, and mechanistic analysis remains rare. Moreover, the requirements for clinicians, lawyers, and financial analysts differ in the evidence they need, and the usefulness of each method in a specific domain heavily depends on whether it exposes that evidence.

A fundamental limitation connects the four families of XAI for LLMs: each produces an artifact whose relation to the model's actual computation is assumed more often than it is tested. The directions identified in Section 6 focus on replacing that assumption with evidence, e.g., interventional validation transferred across approaches, hybrid methods that pair an exposing technique with a validating one, and training objectives that reward faithfulness. Future research on these directions would allow explanations to serve the purpose that motivated this survey: not only to describe what an LLM has generated, but to do so faithfully and to improve the future LLM output.

Acknowledgements This work has been fully supported by ARMADA. The ARMADA project has received funding from Horizon Europe Program for Research and Innovation under the action Horizon-MSCA Doctoral Networks, Grant Agreement No. 101168951 and by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract No 25.00154. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

Declaration of Use of Generative AI. During the preparation of this manuscript, the authors used ChatGPT, Claude Opus 4.6, and Claude Opus 5 to assist with literature discovery, styling figures, as well as for language editing, paraphrasing, and grammar and spelling checks of author-drafted text. All cited works, technical claims, and AI-edited passages were independently verified and integrated by the authors.

References

1. Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2018.
2. Bar Alon, Itamar Zimerman, and Lior Wolf. Faithfulness serum: Mitigating the faithfulness gap in textual explanations of LLM decisions via attribution guidance. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6622–6645. Association for Computational Linguistics, 2026.
3. Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, et al. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025. `https://transformer-circuits.pub/2025/attribution-graphs/methods.html`.
4. Sihem Amer-Yahia, Jasmina Bogojeska, Roberta Facchinetti, Valeria Franceschi, Aristides Gionis, Katja Hose, Georgia Koutrika, Roger Kouyos, Matteo Lissandrini, Silviu Maniu, et al. Towards reliable conversational data analytics. In *2025 EDBT/ICDT 2025 Joint Conference*, 2025.
5. Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. Learning from mistakes makes llm better reasoner, 2024.
6. Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.
7. Shahin Atakishiyev, Housam K. B. Babiker, Jiayi Dai, Nawshad Farruque, Teruaki Hayashi, Nafisa Sadaf Hriti, Md Abed Rahman, Iain Smith, Mi-Young Kim, Osmar R. Zaïane, et al. Explainability of Large Language Models: Opportunities and Challenges toward Generating Trustworthy Explanations, 2025.
8. Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*, pages 722–735. Springer, 2007.
9. Jinheon Baek, Alham Fikri Aji, and Amir Saffari. Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 78–106. ACL, 2023.
10. Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback, 2022.
11. Fazl Barez, Tung-Yu Wu, Iván Arcuschin, Michael Lan, Vincent Wang, Noah Siegel, Nicolas Collignon, Clement Neo, Isabelle Lee, Alasdair Paren, et al. Chain-of-thought is not explainability. *alphaXiv*, abs:2025.02v2, 2025.
12. Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.
13. Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances, 2021.
14. Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety – a review, 2024.
15. Debarun Bhattacharjya, Balaji Ganesan, Michael Glass, Junkyu Lee, Radu Marinescu, Katya Mirylenka, and Xiao Shou. Consistency-based black-box uncertainty quantification for text-to-sql. In *Annual Conference on Neural Information Processing Systems*, 2024.
16. Debarun Bhattacharjya, Balaji Ganesan, Junkyu Lee, Radu Marinescu, Katsiaryna Mirylenka, Michael R. Glass, and Xiao Shou. SIMBA UQ: similarity-based aggregation for uncertainty quantification in large language models. In *EMNLP (Findings)*, pages 15880–15894. Association for Computational Linguistics, 2025.
17. Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`, 2023.
18. Olivier Bodenreider. The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, 32(Database issue):D267–D270, 2004.
19. Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of Artificial General Intelligence: Early experiments with GPT-4, 2023.
20. Zouying Cao, Jiaji Deng, Li Yu, Weikang Zhou, Zhaoyang Liu, Bolin Ding, and Hai Zhao. Remember me, refine me: A dynamic procedural memory framework for experience-driven agent evolution. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Findings of the Association for Computational Linguistics: ACL 2026*, pages 16803–16822, San Diego, California, United States, July 2026. ACL.
21. Jurgita Černevičienė and Audrius Kabašinskas. Explainable artificial intelligence (XAI) in finance: A systematic literature review. *Artificial Intelligence Review*, 57:216, 2024.
22. Sky CH-Wang, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. Do androids know they're only dreaming of electric sheep?, 2024.
23. Lawrence Chan, Adrià Garriga-Alonso, Nicholas Goldowsky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. Causal scrubbing: a method for rigorously testing interpretability hypotheses, 2022. `https://www.alignmentforum.org/posts/JvZhhzycHu2Yd57RN/causal-scrubbing-a-method-for-rigorously-testing`.
24. Yeoktatt Cheah, María Pérez-Ortiz, Noah Y. Siegel, and Oana-Maria Camburu. Training large language models for self-explanation faithfulness. In *ICLR 2026 Workshop on Representational Alignment (Re-Align)*, 2026.
25. Hao Chen, Zhuang Zhao, Ziqi Shuai, and Jifeng Xuan. Testing Explainability of Chain of Thought for Large Language Models. *Applied Sciences*, 16(7):3112, 2026.
26. Qinyu Chen, Wenhao Wu, and Sujian Li. Exploring In-Context Learning for Knowledge Grounded Dialog Generation. In *Findings of the ACL: EMNLP 2023*, pages 10071–10081. ACL, 2023.
27. Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks, 2023.
28. Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger,

et al. Reasoning Models Don't Always Say What They Think. *arXiv*, abs:2505.05410(arXiv:2505.05410), 2025.
29. Yuemin Chen, Feifan Wu, Jingwei Wang, Hao Qian, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, and Meng Wang. Knowledge-augmented financial market analysis and report generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1207–1217, Miami, Florida, US, November 2024. ACL.
30. Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory, 2025.
31. Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. Navigate through Enigmatic Labyrinth A Survey of Chain of Thought Reasoning: Advances, Frontiers and Future. In *Proceedings of the 62nd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 1173–1203. ACL, 2024.
32. Yu-Neng Chuang, Guanchu Wang, Chia-Yuan Chang, Ruixiang Tang, Shaochen Zhong, Fan Yang, Andrew Wen, Mengnan Du, Xuanting Cai, Vladimir Braverman, et al. FaithLM: Towards faithful explanations for large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3802–3824. Association for Computational Linguistics, 2026.
33. Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023.
34. Michele Dallachiesa, Besmira Nushi, Katsiaryna Mirylenka, and Themis Palpanas. Similarity matching for uncertain time series: analytical and experimental comparison. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Querying and Mining Uncertain Spatio-Temporal Data*, pages 8–15, 2011.
35. Arun Das and Paul Rad. Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey, 2020.
36. Shuangyan Deng, Zhongsheng Wang, Rui Mao, Ciprian Doru Giurcăneanu, and Jiamou Liu. CLER: Improving multimodal financial reasoning by cross-MLLM error reflection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 30494–30502, 2026.
37. Aniket Didolkar, Nicolas Ballas, Sanjeev Arora, and Anirudh Goyal. Metacognitive reuse: Turning recurring llm reasoning into concise behaviors, 2025.
38. Tommaso Dolci, Milos Jovanovik, and Katja Hose. Reducing llm hallucinations with knowledge graphs: A survey. *HAL*, hal-05676631, 2026.
39. Tommaso Dolci, Milos Jovanovik, and Katja Hose. Towards LLM-KG Symbiosis for Reducing Factual Hallucinations. In *Proceedings of the Workshops of the EDBT/ICDT Joint Conference*, volume 4192. CEUR-WS, 2026.
40. Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning, 2017.
41. Catalina Dragusin, Katsiaryna Mirylenka, Christoph Miksovic Czasch, Michael Glass, Nahuel Defosse, Paolo Scotton, and Thomas Gschwind. Grounding llms for database exploration: Intent scoping and paraphrasing for robust nl2sql. *Proceedings of the VLDB Endowment*, page 8097, 2025.
42. Rudresh Dwivedi, Devam Dave, Het Naik, Smiti Singhal, Rana Omer, Pankesh Patel, Bin Qian, Zhenyu Wen, Tejal Shah, Graham Morgan, et al. Explainable AI (XAI): Core Ideas, Techniques, and Solutions. *ACM Computing Surveys*, 55(9):194:1–194:33, 2023.
43. Ola El Khatib. Reasoning over incomplete knowledge graphs. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, page 6785–6788, New York, NY, USA, 2025. Association for Computing Machinery.
44. Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition, 2022.
45. Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. RAGAs: Automated Evaluation of Retrieval Augmented Generation. In *Proceedings of the 18th Conference of the European Chapter of the ACL: System Demonstrations*, pages 150–158. ACL, 2024.
46. Jinyuan Fang, Zaiqiao Meng, and Craig MacDonald. TRACE the evidence: Constructing knowledge-grounded reasoning chains for retrieval-augmented generation. In *Findings of the ACL: EMNLP 2024*, pages 8472–8494, Miami, Florida, USA, 2024. ACL.
47. Farima Fatahi Bayat, Kun Qian, Benjamin Han, Yisi Sang, Anton Belyy, Samira Khorshidi, Fei Wu, Ihab Ilyas, and Yunyao Li. FLEEK: Factual Error Detection and Correction with Evidence Retrieved from External Knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 124–130. ACL, 2023.
48. Xueyang Feng, Bo Lan, Quanyu Dai, Lei Wang, Jiakai Tang, Xu Chen, Zhenhua Dong, and Ji-Rong Wen. Improving retrospective language agents via joint policy gradient optimization. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the ACL: Human Language Technologies (Volume 1: Long Papers)*, pages 112–141, Albuquerque, New Mexico, April 2025. ACL.
49. Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. A primer on the inner workings of transformer-based language models, 2024.
50. Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders, 2024.
51. Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore, 2023. ACL.
52. Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, et al. Causal abstraction: A theoretical foundation for mechanistic interpretability, 2025.
53. Hejia Geng and Leo Liu. Parthenon Law: A self-evolving legal-agent framework. *arXiv preprint arXiv:2606.04602*, 2026.
54. GeoNames. The GeoNames geographical database. `https://www.geonames.org/`. Accessed: June 22, 2026.
55. Ashkan Golgoon, Khashayar Filom, and Arjun Ravi Kannan. Mechanistic interpretability of large language models with applications to the financial services industry. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 660–668, 2024.
56. Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large

language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*, 2024.
57. Yu Gu, Jingjing Fu, Xiaodong Liu, Jeya Maria Jose Valanarasu, Noel C. F. Codella, Reuben Tan, Qianchu Liu, Ying Jin, Sheng Zhang, Jinyu Wang, et al. Evaluating the robustness and readiness of large frontier models in health ai applications. *Nature Medicine*, 32(8):3023–3031, 2026.
58. Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.
59. Wes Gurnee and Max Tegmark. Language models represent space and time, 2024.
60. Noah Hampp, Katya Mirylenka, and Michael Glass. Optimizing large language models for robust domain-specific text-to-sql: From prompting to preference alignment. In *Proceedings of the 11th Edition of the Swiss Text Analytics Conference*, pages 63–74, 2026.
61. Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A. Rossi, Subhabrata Mukherjee, Xianfeng Tang, et al. Retrieval-Augmented Generation with Graphs (GraphRAG). *arXiv*, abs:2501.00309, 2025.
62. Michael Hanna, Ollie Liu, and Alexandre Variengien. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model, 2023.
63. Reilly Haskins and Benjamin Adams. KEA Explain: Explanations of Hallucinations using Graph Kernel Analysis. In *Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning*, pages 1041–1058. PMLR, 2025.
64. Bolei He, Nuo Chen, Xinran He, Lingyong Yan, Zhenkai Wei, Jinchang Luo, and Zhen-Hua Ling. Retrieving, rethinking and revising: The chain-of-verification can improve retrieval augmented generation. In *Findings of the ACL: EMNLP 2024*, pages 10371–10393, Miami, Florida, USA, 2024. ACL.
65. Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge Graphs. *ACM Comput. Surv.*, 54(4):71:1–71:37, 2021.
66. Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-STar: Training verifiers for self-taught reasoners. In *First Conference on Language Modeling*, 2024.
67. Xuming Hu, Junzhe Chen, Xiaochuan Li, Yufei Guo, Lijie Wen, Philip S. Yu, and Zhijiang Guo. Towards Understanding Factual Knowledge of Large Language Models. In *The Twelfth International Conference on Learning Representations*, 2023.
68. Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet, 2024.
69. Lei Huang, Xiaocheng Feng, Weitao Ma, Yuxuan Gu, Weihong Zhong, Xiachong Feng, Weijiang Yu, Weihua Peng, Duyu Tang, Dandan Tu, et al. Learning fine-grained grounded citations for attributed large language models. In *Findings of the ACL: ACL 2024*, pages 14095–14113, Bangkok, Thailand, 2024. ACL.
70. Yue Huang, Yanyuan Chen, Dexuan Xu, Weihua Yue, Huamin Zhang, Meikang Qiu, and Yu Huang. MedReflect: Teaching medical LLMs to self-improve via reflective correction. *arXiv preprint arXiv:2510.03687*, 2025.
71. Jeongyeon Hwang, Junyoung Park, Hyejin Park, Dongwoo Kim, Sangdon Park, and Jungseul Ok. Retrieval-augmented generation with estimation of source reliability. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34279–34303, Suzhou, China, 2025. ACL.
72. Alon Jacovi and Yoav Goldberg. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proceedings of the 58th Annual Meeting of the ACL*, pages 4198–4205. ACL, 2020.
73. Minbyul Jeong, Jiwoong Sohn, Mujeen Sung, and Jaewoo Kang. Improving medical reasoning through retrieval and self-reflection with retrieval-augmented large language models. *Bioinformatics*, 40(Supplement_1):i119–i129, 2024.
74. Dongfu Jiang, Jin Zhang, Orion Weller, Nathaniel Weir, Benjamin Van Durme, and Daniel Khashabi. SELF-[IN]CORRECT: LLMs struggle with discriminating self-generated responses. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24266–24275, 2025.
75. Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. StructGPT: A General Framework for Large Language Model to Reason over Structured Data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251. ACL, 2023.
76. Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. A Survey on Large Language Models for Code Generation. *ACM Trans. Softw. Eng. Methodol.*, 35(2):58:1–58:72, 2026.
77. Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the ACL*, 12:1417–1440, 2024.
78. George Katsogiannis-Meimarakis, Katsiaryna Mirylenka, Paolo Scotton, Francesco Fusco, and Abdel Labbi. In-depth analysis of llm-based schema linking. In *EDBT*, pages 117–130, 2026.
79. Joshua Kelsall, Xingwei Tan, Aislinn Bergin, Jiahong Chen, Maria Waheed, Tom Sorell, Rob Procter, Maria Liakata, Jenny Chim, and Serene Chi. A rapid evidence review of evaluation techniques for large language models in legal use cases: Trends, gaps, and recommendations for future research. *AI & Society*, 41(4):4025–4043, 2026.
80. Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large Language Models are Zero-Shot Reasoners. In *Advances in Neural Information Processing Systems*, 2022.
81. Evgeny Krivosheev, Mattia Atzeni, Katsiaryna Mirylenka, Paolo Scotton, and Fabio Casati. Siamese graph neural networks for data integration, 2020.
82. Evgeny Krivosheev, Mattia Atzeni, Katsiaryna Mirylenka, Paolo Scotton, Christoph Miksovic, and Anton Zorin. Business entity matching with siamese graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 16054–16056, 2021.
83. Evgeny Krivosheev, Katsiaryna Mirylenka, Mattia Atzeni, and Paolo Scotton. Graph neural networks for entity matching. In *2023 IEEE International Conference on Big Data (BigData)*, pages 6212–6214, 2023.
84. Michael Krumdick, Rik Koncel-Kedziorski, Viet Dac Lai, Varshini Reddy, Charles Lovering, and Chris Tanner.

Bizbench: A quantitative reasoning benchmark for business and finance. In *Proceedings of the 62nd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 8309–8332, 2024.

85. Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.
86. Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring Faithfulness in Chain-of-Thought Reasoning, 2023.
87. Ernests Lavrinovics, Russa Biswas, Johannes Bjerva, and Katja Hose. Knowledge Graphs, Large Language Models, and Hallucinations: An NLP Perspective. *Journal of Web Semantics*, 85:100844, 2025.
88. Patrick Leask, Bart Bussmann, Michael Pearce, Joseph Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis, 2025.
89. Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 26874–26901. PMLR, 21–27 Jul 2024.
90. Samuel Lewis-Lim, Xingwei Tan, Zhixue Zhao, and Nikolaos Aletras. Analysing Chain of Thought Dynamics: Active Guidance or Unfaithful Post-hoc Rationalisation? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29838–29853. ACL, 2025.
91. Haitao Li, Yifan Chen, Yiran Hu, Qingyao Ai, Junjie Chen, Xiaoyu Yang, Jianhui Yang, Yueyue Wu, Zeyang Liu, and Yiqun Liu. LexRAG: Benchmarking retrieval-augmented generation in multi-turn legal consultation conversation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3606–3615. Association for Computing Machinery, 2025.
92. Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024.
93. Charlie Lindsay, Francesca Toni, Tien Dat Nguyen, Tan Minh Nguyen, Ngoc Anh Trang Pham, Quang Huy Chu, Hoang Nhat Nguyen, and Le-Minh Nguyen. Argumentative LLMs for legal information entailment. In *Legal Knowledge and Information Systems: JURIX 2025: The Thirty-eighth Annual Conference*, volume 416 of *Frontiers in Artificial Intelligence and Applications*, pages 121–132. IOS Press, 2025.
94. Zachary C. Lipton. The mythos of model interpretability. *Communications of the ACM*, 61(10):36–43, 2018.
95. Guangyi Liu, Yongqi Zhang, Yong Li, and Quanming Yao. Dual Reasoning: A GNN-LLM Collaborative Framework for Knowledge Graph Question Answering. In *The Second Conference on Parsimony and Learning (Proceedings Track)*, 2025.
96. Qian Liu, Hang Yu, Qiqi Wang, Qi Xu, Jinpeng Li, Zhuoqun Zou, Rui Mao, and Erik Cambria. Legal knowledge infusion for large language models: A survey. *Information Fusion*, 125:103426, 2026.
97. Siru Liu, Allison B McCoy, and Adam Wright. Improving large language model applications in biomedicine with retrieval-augmented generation: a systematic review, meta-analysis, and clinical development guidelines. *Journal of the American Medical Informatics Association*, 32(4):605–615, 2025.
98. Yitao Liu, Chenglei Si, Karthik R Narasimhan, and Shunyu Yao. Contextual experience replay for self-improvement of language agents. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 14179–14198, Vienna, Austria, July 2025. ACL.
99. Ivan Lopez, Akshay Swaminathan, Karthik S. Vedula, Sanjana Narayanan, Fateme Nateghi Haredasht, Stephen P. Ma, April S. Liang, Steven Tate, Manoj Maddali, Robert Gallo, et al. Clinical entity augmented retrieval for clinical information extraction. *npj Digital Medicine*, 8(1):45, 2025.
100. Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.
101. Haitong Luo, Fali Wang, Weiyao Zhang, Xianren Zhang, Zhiwei Zhang, Tianxiang Zhao, Minhua Lin, Jiahao Zhang, Hui Liu, Xianfeng Tang, et al. Graph-Assisted Large Language Models: A Perspective on Mitigating Intrinsic Limitations. In *Findings of the ACL: ACL 2026*, pages 18936–18955. ACL, 2026.
102. Haoyan Luo and Lucia Specia. From Understanding to Utilization: A Survey on Explainability for Large Language Models, 2024.
103. Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In *The Twelfth International Conference on Learning Representations*, 2023.
104. Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuan-Fang Li, Chen Gong, and Shirui Pan. Graph-constrained Reasoning: Faithful Reasoning on Knowledge Graphs with Large Language Models. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 41540–41565. PMLR, 2025.
105. Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
106. Aman Madaan and Amir Yazdanbakhsh. Text and Patterns: For Effective Chain of Thought, It Takes Two to Tango, 2022.
107. Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. Clin: A continually learning language agent for rapid task adaptation and generalization, 2023.
108. Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. Llm critics help catch llm bugs, 2024.

109. Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023.
110. Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one?, 2019.
111. Katsiaryna Mirylenka, Themis Palpanas, Graham Cormode, and Divesh Srivastava. Finding interesting correlations with conditional heavy hitters. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 1069–1080, 2013.
112. Katya Mirylenka, Egor Malykh, Mahdyar Ravanbakhsh, Michael Gygli, Marco-Andrea Buchmann, Andrew Dzhoha, Svitlana Borzenko, Francesca Catino, Mohamed Gaafar, Thomas Kober, et al. From clicks to clips: A multimodal retrieval system for e-commerce video recommendations. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track, Demo Track and Industrial Track*, 2026.
113. Katya Mirylenka, Egor Malykh, Mahdyar Ravanbakhsh, Michael Gygli, Marco-Andrea Buchmann, Andrew Dzhoha, Svitlana Borzenko, Francesca Catino, Mohamed Gaafar, Maarten Versteegh, et al. Vcg: A multimodal retrieval framework for e-commerce video feeds under extreme cold-start conditions, 2026.
114. Venkatesh Mishra, Bimsara Pathiraja, Mihir Parmar, Sat Chidananda, Jayanth Srinivasa, Gaowen Liu, Ali Payani, and Chitta Baral. Investigating the shortcomings of LLMs in step-by-step legal reasoning. In *Findings of the ACL: NAACL 2025*, pages 7810–7841, Albuquerque, New Mexico, April 2025. ACL.
115. Giovanni Monea, Maxime Peyrard, Martin Josifoski, Vishrav Chaudhary, Jason Eisner, Emre Kıcıman, Hamid Palangi, Barun Patra, and Robert West. A glitch in the matrix? locating and detecting language model grounding with fakepedia, 2024.
116. Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, et al. The quest for the right mediator: Surveying mechanistic interpretability through the lens of causal mediation analysis, 2025.
117. Maxime Méloux, Giada Dirupo, François Portet, and Maxime Peyrard. The dead salmons of ai interpretability, 2025.
118. Maxime Méloux, Tiago Pimentel, François Portet, and Maxime Peyrard. Validating causal abstraction metrics on simulated complex systems, 2026.
119. Neel Nanda. Attribution patching: Activation patching at industrial scale, 2023. `https://www.neelnanda.io/mechanistic-interpretability/attribution-patching`.
120. Usman Naseem. Mechanistic interpretability for large language model alignment: Progress, challenges, and future directions, 2026.
121. Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models. In *Proceedings of the 62nd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 10862–10878. ACL, 2024.
122. OpenAI, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, et al. OpenAI o1 System Card, 2026.
123. Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory, 2026.
124. Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, et al. Large Language Models and Knowledge Graphs: Opportunities and Challenges. *Transactions on Graph Data and Knowledge*, 1(1):2:1–2:38, 2023.
125. Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599, 2024.
126. Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery.
127. Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. Graph Retrieval-Augmented Generation: A Survey. *ACM Trans. Inf. Syst.*, 44(2):35:1–35:52, 2025.
128. Riccardo Pozzi, Matteo Palmonari, Andrea Coletta, Luigi Bellomarini, Jens Lehmann, and Sahar Vahdati. ReFactX: Scalable Reasoning with Reliable Facts via Constrained Generation. In *The Semantic Web – ISWC 2025*, pages 290–308. Springer Nature Switzerland, 2026.
129. Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968, Singapore, December 2023. ACL.
130. Jirui Qi, Gabriele Sarti, Raquel Fernández, and Arianna Bisazza. Model internals-based answer attribution for trustworthy retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6037–6053, Miami, Florida, USA, 2024. ACL.
131. Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, et al. Question Decomposition Improves the Faithfulness of Model-Generated Reasoning, 2023.
132. Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. A practical review of mechanistic interpretability for transformer-based language models, 2025.
133. Mohamed Rashad, Ahmed Zahran, Abanoub Amin, Amr Abdelaal, and Mohamed Altantawy. FactAlign: Fact-Level Hallucination Detection and Classification Through Knowledge Graph Alignment. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pages 79–84. ACL, 2024.
134. Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840, 2023.
135. Matthew Renze and Erhan Guven. Self-reflection in large language model agents: Effects on problem-solving performance. In *2024 2nd International Conference on*

Foundation and Large Language Models (FLLM), pages 516–525. IEEE, 2024.

136. Carl O. Retzlaff, Alessa Angerschmid, Anna Saranti, David Schneeberger, Richard Röttger, Heimo Müller, and Andreas Holzinger. Post-hoc vs ante-hoc explanations: xai design guidelines for data scientists. *Cogn. Syst. Res.*, 86(C), August 2024.
137. Leonardo F. R. Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, and Mohit Bansal. FactGraph: Evaluating Factuality in Summarization with Semantic Graph Representations. In *Proceedings of the 2022 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 3238–3253. ACL, 2022.
138. Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.
139. Rikard Rosenbacke, Åsa Melhus, Martin McKee, and David Stuckler. How explainable artificial intelligence can increase or decrease clinicians' trust in ai applications in health care: systematic review. *Jmir Ai*, 3:e53207, 2024.
140. Hannah Sansford, Nicholas Richardson, Hermina Petric Maretic, and Juba Nait Saada. GraphEval: A Knowledge-Graph Based LLM Hallucination Evaluation Framework. In *KiL@KDD*, volume 3894 of *CEUR Workshop Proceedings*, pages 20–31. CEUR-WS.org, 2024.
141. William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators, 2022.
142. Priyanka Sen, Sandeep Mavadia, and Amir Saffari. Knowledge Graph-augmented Language Models for Complex Question Answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 1–8. ACL, 2023.
143. Sofia Serrano and Noah A. Smith. Is Attention Interpretable? In *Proceedings of the 57th Annual Meeting of the ACL*, pages 2931–2951. ACL, 2019.
144. Sergio Servantez, Joe Barrow, Kristian Hammond, and Rajiv Jain. Chain of logic: Rule-based reasoning with large language models. In *Findings of the ACL: ACL 2024*, pages 2721–2733, Bangkok, Thailand, 2024. ACL.
145. Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, et al. Open problems in mechanistic interpretability, 2025.
146. Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
147. Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity. *arXiv*, abs:2506.06941(arXiv:2506.06941), 2025.
148. Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024.
149. Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. Rethinking Interpretability in the Era of Large Language Models, 2024.
150. Aviv Slobodkin, Eran Hirsch, Arie Cattan, Tal Schuster, and Ido Dagan. Attribute first, then generate: Locally-attributable grounded text generation. In *Proceedings of the 62nd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 3309–3344, Bangkok, Thailand, 2024. ACL.
151. Jiwoong Sohn, Yein Park, Chanwoong Yoon, Sihyeon Park, Hyeon Hwang, Mujeen Sung, Hyunjae Kim, and Jaewoo Kang. Rationale-guided retrieval augmented generation for medical question answering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the ACL: Human Language Technologies (Volume 1: Long Papers)*, pages 12739–12753, 2025.
152. Yurun Song, Xiangqing Shen, and Rui Xia. From Phrases to Subgraphs: Fine-Grained Semantic Parsing for Knowledge Graph Question Answering. In *Findings of the ACL: ACL 2025*, pages 5232–5246. ACL, 2025.
153. Yurun Song, Xiangqing Shen, Jianfei Yu, and Rui Xia. Interactive Semantic Parsing with Reinforcement Learning for Knowledge Graph Reasoning. In *Findings of the ACL: ACL 2026*, pages 20464–20484. ACL, 2026.
154. Robyn Speer, Joshua Chin, and Catherine Havasi. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), 2017.
155. Zhaoli Su, Hong Song, Yucong Lin, You Wu, Xutao Weng, Zhongxuan Mao, Bowen Liu, Hongxia Yin, and Jian Yang. MedKit: Multi-level feature distillation with knowledge injection for radiology report generation. *Expert Systems with Applications*, 296:129003, 2026.
156. Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *The Twelfth International Conference on Learning Representations*, 2023.
157. Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory, 2025.
158. Zhen Tan, Hanning Xue, Yixin Wan, Rui Xiong, Xiaokang Chu, Xiang Li, and Jiawei Liu. Efficient transcoder adaptation for fine-tuned models: Revealing medical reasoning mechanisms in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33196–33204, 2026.
159. Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, et al. Agent kb: Leveraging cross-domain experience for agentic problem solving, 2025.
160. Zichen Tang, E Haihong, Ziyan Ma, Haoyang He, Jiacheng Liu, Zhongjun Yang, Zihua Rong, Rongjin Li, Kun Ji, Qing Huang, et al. Financereasoning: Benchmarking financial numerical reasoning more credible, comprehensive and challenging. In *Proceedings of the 63rd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 15721–15749, 2025.
161. Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, et al. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. `https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html`.

162. Shiyu Tian, Shuyue Xing, Xingrui Li, Yangyang Luo, Caixia Yuan, Wei Chen, Huixing Jiang, and Xiaojie Wang. A Systematic Exploration of Knowledge Graph Alignment with Large Language Models in Retrieval Augmented Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(24):25291–25299, 2025.
163. Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In *Proceedings of the 61st Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada, 2023. ACL.
164. Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. In *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.
165. Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, 2014.
166. Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024.
167. Jianing Wang, Qiushi Sun, Xiang Li, and Ming Gao. Boosting Language Models Reasoning with Chain-of-Knowledge Prompting. In *Proceedings of the 62nd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 4958–4981. ACL, 2024.
168. Jianing Wang, Qiushi Sun, Xiang Li, and Ming Gao. Boosting language models reasoning with chain-of-knowledge prompting. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 4958–4981, Bangkok, Thailand, August 2024. ACL.
169. Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small, 2022.
170. Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand, August 2024. ACL.
171. Tianlu Wang, Ping Yu, Xiaoqing Ellen Tan, Sean O'Brien, Ramakanth Pasunuru, Jane Dwivedi-Yu, Olga Golovneva, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. Shepherd: A critic for language model generation, 2023.
172. Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*, 2022.
173. Yubo Wang, Xiang Yue, and Wenhu Chen. Critique fine-tuning: Learning to critique is more effective than learning to imitate. In *Second Conference on Language Modeling*, 2025.
174. Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 63897–63911. PMLR, 13–19 Jul 2025.
175. Patrick Weber, K. Valerie Carl, and Oliver Hinz. Applications of explainable artificial intelligence in finance—a systematic review of finance, information systems, and computer science literature. *Management Review Quarterly*, 74(2):867–907, 2024.
176. Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, 2022.
177. Yeo Wei Jie, Ranjan Satapathy, Rick Goh, and Erik Cambria. How interpretable are reasoning explanations from prompting large language models? In *Findings of the ACL: NAACL 2024*, page 2148–2164. ACL, 2024.
178. Yeo Wei Jie, Ranjan Satapathy, Rick Goh, and Erik Cambria. How Interpretable are Reasoning Explanations from Prompting Large Language Models? In *Findings of the ACL: NAACL 2024*, pages 2148–2164. ACL, 2024.
179. Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*, 2023.
180. Yilin Wen, Zifeng Wang, and Jimeng Sun. MindMap: Knowledge Graph Prompting Sparks Graph of Thoughts in Large Language Models. In *Proceedings of the 62nd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 10370–10388. ACL, 2024.
181. Lukas Wertz, Jasmina Bogojeska, Katsiaryna Mirylenka, and Jonas Kuhn. Reinforced active learning for low-resource, domain-specific, multi-label text classification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10959–10977, 2023.
182. Lukas Wertz, Katsiaryna Mirylenka, Jonas Kuhn, and Jasmina Bogojeska. Investigating active learning sampling strategies for extreme multi label text classification. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4597–4605, 2022.
183. Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, Yueming Jin, and Vicente Grau. Medical graph RAG: Evidence-based medical large language model via graph retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 28443–28467, Vienna, Austria, 2025. ACL.
184. Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason E Weston, and Sainbayar Sukhbaatar. Meta-rewarding language models: Self-improving alignment with LLM-as-a-meta-judge. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11537–11554, Suzhou, China, November 2025. ACL.
185. Yike Wu, Nan Hu, Sheng Bi, Guilin Qi, Jie Ren, Anhuan Xie, and Wei Song. Retrieve-Rewrite-Answer: A KG-to-Text Enhanced LLMs Framework for Knowledge Graph Question Answering, 2023.
186. Quan Xiao, Debarun Bhattacharjya, Balaji Ganesan, Radu Marinescu, Katsiaryna Mirylenka, Nhan H. Pham, Michael Glass, and Junkyu Lee. The consistency hypothesis in uncertainty quantification for large language

models. In *Proceedings of the Forty-First Conference on Uncertainty in Artificial Intelligence (UAI 2025)*, 2025.
187. Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. Pride and prejudice: LLM amplifies self-bias in self-refinement. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 15474–15492, Bangkok, Thailand, August 2024. ACL.
188. Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents, 2025.
189. Xinyu Yang, Chenlong Deng, Tongyu Wen, Binyu Xie, and Zhicheng Dou. LawThinker: A deep research legal agent in dynamic environments. *arXiv preprint arXiv:2602.12056*, 2026.
190. Zhuoyi Yang and Liyue Shen. Tempa-vlp: Temporal-aware vision-language pretraining for longitudinal exploration in chest x-ray image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4625–4634, 2025.
191. Zongxian Yang, Jiayu Qian, Zegao Peng, Haoyu Zhang, Yu-An Huang, K. C. Tan, and Zhi-An Huang. MedREFL: Medical reasoning enhancement via self-corrected fine-grained reflection. *arXiv preprint arXiv:2506.13793*, 2025.
192. Rujing Yao, Yang Wu, Chenghao Wang, Jingwei Xiong, Fang Wang, and Xiaozhong Liu. Elevating legal llm responses: harnessing trainable logical structures and semantic knowledge with legal reasoning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the ACL: Human Language Technologies (Volume 1: Long Papers)*, pages 5630–5642, 2025.
193. Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822, 2023.
194. Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh R N, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, et al. Retroformer: Retrospective large language agents with policy gradient optimization. In *The Twelfth International Conference on Learning Representations*, 2024.
195. Yao Yao, Zuchao Li, and Hai Zhao. GoT: Effective Graph-of-Thought Reasoning in Language Models. In *Findings of the ACL: NAACL 2024*, pages 2901–2921. ACL, 2024.
196. Zonghai Yao, Aditya Parashar, Huixue Zhou, Won Seok Jang, Feiyun Ouyang, Zhichao Yang, and Hong Yu. MCQG-SRefine: Multiple choice question generation and evaluation with iterative self-critique, correction, and comparison feedback. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the ACL: Human Language Technologies (Volume 1: Long Papers)*, pages 10728–10777, Albuquerque, New Mexico, 2025. ACL.
197. Jiong Yu, Sixing Wu, Jiahao Chen, and Wei Zhou. LLMs as Collaborator: Demands-Guided Collaborative Retrieval-Augmented Generation for Commonsense Knowledge-Grounded Open-Domain Dialogue Systems. In *Findings of the ACL: EMNLP 2024*, pages 13586–13612. ACL, 2024.
198. Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan W. Suchow, Zhenyu Cui, Rong Liu, et al. FinCon: A synthesized LLM multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc., 2024.
199. Yue Yu, Zhengxing Chen, Aston Zhang, Liang Tan, Chenguang Zhu, Richard Yuanzhe Pang, Yundi Qian, Xuewei Wang, Suchin Gururangan, Chao Zhang, et al. Self-generated critiques boost reward modeling for language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the ACL: Human Language Technologies (Volume 1: Long Papers)*, pages 11499–11514, Albuquerque, New Mexico, April 2025. ACL.
200. Weikang Yuan, Junjie Cao, Zhuoren Jiang, Yangyang Kang, Jun Lin, Kaisong Song, Tianqianjin Lin, Pengwei Yan, Changlong Sun, and Xiaozhong Liu. Can large language models grasp legal theories? enhance legal reasoning with insights from multi-agent collaboration. In *Findings of the ACL: EMNLP 2024*, pages 7577–7597, 2024.
201. Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 57905–57923. PMLR, 21–27 Jul 2024.
202. Mert Yuksekgonul, Federico Bianchi, James Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative AI by backpropagating language model feedback. *Nature*, 639(8055):609–616, 2025.
203. Hannah Yun, Junyeong Maeng, Eunsong Kang, and Heung-Il Suk. Diff-rrg: Longitudinal disease-wise patch difference as guidance for llm-based radiology report generation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2025*, volume 15966 of *Lecture Notes in Computer Science*, pages 152–161. Springer Nature Switzerland, 2025.
204. Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: self-taught reasoner bootstrapping reasoning with reasoning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.
205. Hengyuan Zhang, Zhihao Zhang, Mingyang Wang, Zunhai Su, Yiwei Wang, Qianli Wang, Shuzhou Yuan, Ercong Nie, Xufeng Duan, Feijiang Han, et al. Locate, steer, and improve: A practical survey of actionable mechanistic interpretability in large language models, 2026.
206. Jiawei Zhang. Graph-ToolFormer: To Empower LLMs with Graph Reasoning Ability via Prompt Augmented by ChatGPT, 2023.
207. Kepu Zhang, Weijie Yu, Sunhao Dai, and Jun Xu. Citalaw: Enhancing llm with citations in legal domain. In *Findings of the ACL: ACL 2025*, pages 11183–11196, 2025.
208. Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic context engineering: Evolving contexts for self-improving language models, 2026.
209. Xue Zhang. Constitution or collapse? exploring constitutional ai with llama 3-8b, 2025.
210. Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI*

Conference on Artificial Intelligence, volume 38, pages 19632–19642, 2024.
211. Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for Large Language Models: A Survey. *ACM Trans. Intell. Syst. Technol.*, 15(2):20:1–20:38, 2024.
212. Ruilin Zhao, Feng Zhao, and Hong Zhang. Correcting on Graph: Faithful Semantic Parsing over Knowledge Graphs with Large Language Models. In *Findings of the ACL: ACL 2025*, pages 5364–5376. ACL, 2025.
213. Suifeng Zhao, Zhuoran Jin, Sujian Li, and Jun Gao. Finragbench-v: A benchmark for multimodal rag with visual citation in the financial domain. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4215–4249, 2025.
214. Xuejiao Zhao, Siyan Liu, Su-Yin Yang, and Chunyan Miao. MedRAG: Enhancing retrieval-augmented generation with knowledge graph-elicited reasoning for healthcare copilot. In *Proceedings of the ACM on Web Conference 2025*, pages 4442–4457. ACM, 2025.
215. Yilun Zhao, Yitao Long, Tintin Jiang, Chengye Wang, Weiyuan Chen, Hongjun Liu, Xiangru Tang, Yiming Zhang, Chen Zhao, and Arman Cohan. Findver: Explainable claim verification over long and hybrid-content financial documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14739–14752, 2024.
216. Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. Synapse: Trajectory-as-exemplar prompting with memory for computer control. In *The Twelfth International Conference on Learning Representations*, 2024.
217. Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. Knowledge Graph-Guided Retrieval Augmented Generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the ACL: Human Language Technologies (Volume 1: Long Papers)*, pages 8912–8924. ACL, 2025.
218. Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. Large Language Models for Information Retrieval: A Survey. *ACM Trans. Inf. Syst.*, 44(1):12:1–12:54, 2025.