Survey on Mechanistic Interpretability of Reasoning
A Review of Recent Advances

Narges Javid *narges.javid1377@sharif.edu*
Department of Computer Engineering
Sharif University of Technology

Ali Karimi *ali.karimi.32@sharif.edu*
Department of Computer Engineering
Sharif University of Technology

Zeinab Ehyaei *zeinab.ehyaei.00@sharif.edu*
Department of Computer Engineering
Sharif University of Technology

Mohammad Mostafa Rostamkhani *mohammadmostafarostamkhani@gmail.com*
Department of Computer Engineering
Sharif University of Technology

Moein Salimi *s.salimi.moein@gmail.com*
Department of Computer Engineering
Sharif University of Technology

Shaygan Adim *sh83adim@gmail.com*
Department of Computer Engineering
Sharif University of Technology

Abstract

Mechanistic interpretability aims to explain how neural networks carry out computations internally. Reasoning is an important setting for this goal because it often depends on multiple steps, intermediate states, and interactions across layers and tokens that are not visible in the final answer. As language models are increasingly used for complex reasoning, understanding these internal processes is important for explaining how their reasoning works and where it may fail.

This survey reviews recent work on the mechanistic interpretability of reasoning in transformer models, covering controlled reasoning tasks, pretrained language models, and Chain-of-Thought or reasoning-tuned settings. We compare the mechanisms identified by each study, the evidence supporting those explanations, and the conditions under which they continue to hold. Controlled tasks often allow researchers to reconstruct a reasoning process in more detail, while studies of pretrained language models more often identify important components or pathways without fully explaining how they work together. Overall, the literature shows clear progress in finding reasoning-related structure, but less progress in building explanations that reliably predict when a mechanism will generalize or fail.

1 Introduction

Transformer models can solve tasks that involve arithmetic, deduction, search, state tracking, and multi-step composition (Stolfo et al., 2023; Maltoni & Ferrara, 2026; Saparov et al., 2025; Zhang et al., 2025b; Dutta et al., 2024). A correct answer, however, does not tell us how the model reached it. Mechanistic interpretability

(MI) tries to explain this internal process by asking which components carry relevant information, how that information moves through the network, and which computations are needed to produce the answer.

Reasoning is an important setting for MI because it often depends on intermediate states and repeated computation that are not visible in the final output. In controlled tasks, researchers often know the target computation, so they can compare the model's internal process with a known algorithm (Saparov et al., 2025; Zhang et al., 2025b). In pretrained language models, this is harder because reasoning is mixed with stored knowledge, language patterns, and redundant pathways. Chain-of-Thought (CoT) adds another distinction: generated intermediate tokens can provide extra computational steps, but the visible text does not necessarily reveal every hidden computation used to produce the answer (Li et al., 2023; Dutta et al., 2024).

A clear difference across the reviewed papers is how much of the reasoning process they can explain. Controlled studies often reconstruct a relatively complete algorithm or test where it stops working. Studies of pretrained or reasoning-tuned models more often identify an important component, pathway, feature, or timing effect without reconstructing the full process. This is not a ranking of paper quality. Full reconstruction is simply easier when the task and state space are tightly controlled. The distinction is central to this survey: finding a reasoning-related component or representation is useful, but it is different from explaining how the full computation works and predicting where it may fail.

Several papers can also be compared using common terms such as retrieval, routing, state updating, and decision-making (Stolfo et al., 2023; Hong et al., 2025; Cabannes et al., 2024; Biran et al., 2024). These terms are only a way to compare papers, not a general architecture of reasoning. Distributed Fourier computation, task-specific grokked circuits, graph reachability, and prompt-dependent in-context algebra show cases where a simple staged description does not fit well (Nanda et al., 2023; Wang et al., 2024; Saparov et al., 2025; Todd et al., 2026).

To compare the papers consistently, we ask four questions:

1. **What kind of reasoning is studied?** Is it a controlled algorithmic task, reasoning inside a pretrained model, or explicit multi-step generation such as CoT?
2. **What part of the model is being explained?** Is it a component, circuit, feature, token trajectory, internal state, or algorithm?
3. **What evidence supports the explanation?** Does the paper locate information, intervene on it, test whether it is necessary or sufficient, reconstruct a computation, or try to falsify the interpretation?
4. **Where does the explanation still work?** Is it tested on new inputs, longer sequences, different tasks, different distributions, other models, or larger scales?

Using these questions, the survey does three things. First, it separates different strengths of mechanistic evidence instead of treating every interpretability result as equivalent. Second, it compares patterns that appear across multiple tasks while also showing where those patterns stop being useful. Third, it uses limits and disagreements in the reviewed papers to identify a small set of concrete questions for future work.

The rest of the survey is organized as follows. Section 2 introduces the background concepts needed to discuss reasoning mechanisms. Section 3 reviews the main interpretability methods and explains how we compare different kinds of evidence. Section 4 reviews findings across controlled tasks, pretrained language models, explicit CoT, and sparse features. Section 5 compares the studies directly, focusing on how far their explanations go, which patterns recur, and where those explanations break down. Section 6 turns the main disagreements and limits into open questions for future work. Section 7 summarizes the limitations of this survey, and Section 8 concludes.

2 Background

2.1 From model behavior to mechanism

Behavioral performance tells us whether a model solves a task, but not how it does so. Mechanistic interpretability therefore looks inside the model and asks how internal computation produces the observed behavior. Finding task-relevant information in an activation or component is an important first step, but

it does not by itself show that the model actually uses that information. A stronger explanation connects internal representations and components to the computation that produces the answer, ideally with causal interventions or predictions about intermediate states and behavior.

Different forms of evidence answer different questions. A probe may show that information is available at a particular layer, while an intervention can test whether changing that information affects the output. Even then, changing one component does not necessarily reveal the full mechanism, because the model may have redundant or alternative pathways (Heimersheim & Nanda, 2024). Many studies therefore describe smaller circuits made of attention heads, MLPs, residual-stream states, or learned features. The aim is not only to find important components, but to understand how they work together.

2.2 Reasoning in hidden states and generated tokens

Reasoning may happen mainly in hidden activations, or generated intermediate tokens may take part in the computation. We use *latent reasoning* for multi-step processing carried mainly through hidden states before an answer is produced. We use *explicit reasoning* for settings in which intermediate tokens are part of the computation, as in Chain-of-Thought (CoT). These terms describe where intermediate computation happens; they do not imply that visible CoT is a complete explanation of the model's internal reasoning.

Some papers use more specific terms. Maltoni & Ferrara (2026) distinguish *vertical reasoning*, performed through model depth, from *horizontal reasoning*, performed across generated tokens. We keep these as paper-specific terms rather than treating them as general synonyms for latent and explicit reasoning. More detailed distinctions, such as the CoT-I and CoT-I/O settings of Li et al. (2023), are introduced later with the corresponding experiments.

3 Methods and Comparison Framework

3.1 Methods for studying reasoning mechanisms

3.1.1 Causal interventions and circuit discovery

Activation patching replaces internal activations between related runs and measures how behavior changes. Denoising and noising answer different causal questions, and the result depends on the corruption, intervention site, and evaluation metric (Heimersheim & Nanda, 2024). A further caution is that an intervention may push the model into an internal state that would not occur during normal inference. Grant et al. (2026) show that some interventions can activate divergent or otherwise dormant pathways. Patching therefore gives causal evidence under a particular intervention; it does not automatically reveal the model's normal computation.

Automated circuit-discovery methods reduce the amount of manual search, but they use different signals. Table 1 summarizes the main differences. ACDC prunes edges using activation-patching effects (Conmy et al., 2023); CD-T uses contextual-decomposition contributions (Hsu et al., 2025); Query Circuits searches for sparse circuits that preserve a particular query/response behavior (Wu & Barez, 2026); and SICAF combines circuit extraction with token-level influence through depth (Zhang et al., 2025a). Their outputs are not interchangeable: a large contribution is not the same as causal necessity, and a circuit that works for one query is not automatically a reusable mechanism for a whole capability.

Table 1: Automated circuit-analysis methods and the type of evidence they provide. The methods were not evaluated under one common protocol, so this is not a performance ranking.

Method	Main signal	What it directly supports	Main caution
ACDC (Conmy et al., 2023)	Activation-patching effect during edge pruning	A sparse subgraph under a chosen corruption and metric	Redundancy and corruption/metric choices can change which edges appear important.
CD-T (Hsu et al., 2025)	Contextual decomposition contribution	A fine-grained contribution-based circuit	Contribution is not the same as causal necessity.
Query Circuits (Wu & Barez, 2026)	Query-level behavior preserved by a sparse circuit	A sparse explanation for a particular query/response	Local behavior does not establish a stable capability-level circuit.
SICAF (Zhang et al., 2025a)	Circuit extraction plus token influence through depth	A sparse pathway together with a temporal influence account	The interpretation depends on the quality and task coverage of the extracted circuit.

3.1.2 Probing, reconstruction, and sparse features

Probing and representation analysis can show where task-relevant structure becomes recoverable, as in Hou et al. (2023), but decoding alone does not show that the model uses that information causally. Controlled tasks can go further because expected state transitions or algorithms are known and can be compared directly with the model (Nanda et al., 2023; Saparov et al., 2025; Zhang et al., 2025b; Cabannes et al., 2024). A reconstruction is stronger when it predicts intermediate states and continues to work under interventions or boundary tests.

Sparse autoencoders (SAEs) provide another unit of analysis by learning sparse directions from activations. Cunningham et al. (2024) motivate SAEs as a way to obtain more interpretable features than raw neurons, and Marks et al. (2025) connect such features into editable causal graphs. For reasoning, however, the meaning of a feature can be difficult to establish. Steering and patching may show that a feature matters causally, while lexical or stylistic confounds may still make its semantic label misleading (Galichin et al., 2026; Chen et al., 2026; Ma et al., 2026).

3.2 How we compare the evidence

The methods above produce different kinds of evidence. We use five non-cumulative categories to compare the reviewed studies:

1. **Localization/decoding:** task-relevant information is found in a component, feature, or position.
2. **Causal intervention:** changing that object changes behavior in the predicted direction.
3. **Necessity/sufficiency:** the object is tested under removal, restoration, or both.
4. **Mechanistic reconstruction:** components are combined into an explanation that predicts intermediate computation.
5. **Stress testing/falsification:** the explanation is challenged with transfer tests, distribution shifts, counterexamples, paraphrases, or adversarial controls.

These categories form an *evidence profile*, not a score or strict ranking. A study may, for example, provide localization, intervention, and falsification evidence without reconstructing a full circuit.

When comparing papers, we also separate three kinds of claims. A *paper-specific finding* comes directly from one study. A *repeated pattern* appears in several studies. A broader *cross-paper interpretation* is our comparison of the reviewed findings and is presented as an interpretation, not as an experimental result from any one paper.

4 Findings Across Reasoning Settings

This section groups empirical studies by reasoning setting for readability. Within each group, we focus on three questions: what mechanism the paper identifies, what evidence supports it, and what the comparison with nearby studies adds.

4.1 Controlled reasoning tasks

Controlled tasks are problems whose rules, intermediate states, or target computation are specified precisely enough that researchers can compare the model's internal computation with a known reference. The studies in this subsection use that control in different ways: some reconstruct circuits that emerge with generalization, some test whether a learned strategy transfers beyond the setting in which it was identified, and others track how intermediate state is represented and updated during multi-step computation. Together, they show the range of mechanistic questions that become easier to study when the task itself is well specified.

4.1.1 Grokking and generalizing circuits

These papers move from observing generalization to asking what internal mechanism produces it. Power et al. (2022) introduced grokking as delayed generalization: a model first fits the training set and only later begins to generalize. Nanda et al. (2023) then study modular addition mechanistically and recover a Fourier-based circuit. Their analysis shows that the behavioral transition is preceded by gradual internal changes involving memorization, circuit formation, and later cleanup. Wang et al. (2024) extend the question to implicit comparison and composition. Both tasks develop generalizing circuits, but their out-of-distribution (OOD) behavior differs. Taken together, these studies show that identifying a clear internal mechanism can explain how generalization emerges in one setting without guaranteeing the same kind of extrapolation in another.

4.1.2 In-context algebra and graph search

These studies ask whether systematic-looking behavior is produced by one stable algorithm. Todd et al. (2026) study algebraic sequences in which token meanings change with context. The models can generalize to unseen algebraic groups, but targeted distributions and causal tests reveal several strategies, including copying, identity recognition, and cancellation. The mechanism therefore changes with prompt structure rather than following one fixed strategy.

Saparov et al. (2025) study graph connectivity as a controlled search problem and recover a more stable iterative mechanism: the transformer represents reachable vertices and expands that information across layers. However, performance still degrades on larger graphs. Together, the two papers show that systematic behavior can arise either from a relatively stable algorithm or from several prompt-dependent strategies, and that interpretability alone does not guarantee broad generalization.

4.1.3 Tracking and updating intermediate states

A related group of studies asks how intermediate state is carried through repeated reasoning steps. Zhang et al. (2025b) study Transformer+CoT models on state-tracking tasks. They identify late-layer MLP neurons associated with states of an implicit finite-state automaton and test skipped steps, noise, and longer sequences. Because the task state is known, internal activations can be compared directly with the target automaton.

Cabannes et al. (2024) analyze repeated computation through an "iteration head." Attention routes the relevant input while another computation updates the current state, and transfer experiments help separate reusable routing from task-specific update rules. Maltoni & Ferrara (2026) compare token-based horizontal deduction with depth-based vertical deduction in small GPT-style models and show that CoT supervision can help train later vertical inference. Together, these papers show that multi-step reasoning can maintain and update state in different ways: through internal state representations, reusable routing-and-update operations, or additional computation across generated tokens.

4.1.4 What CoT adds in a controlled setting

Li et al. (2023) analyze compositional in-context learning of multi-layer perceptrons and compare ordinary ICL with two step-augmented settings: CoT-I, where intermediate states are provided in the demonstrations, and CoT-I/O, where intermediate outputs are also generated and reused during prediction. Their key decomposition separates an attention-based filtering step from a simpler single-step learning operation. Under the paper's assumptions, CoT-I/O learns an MLP with input dimension d and k neurons using $O(\max(d, k))$

in-context samples, compared with an $\Omega(kd)$ lower bound without step-augmented prompts. They also construct deep linear MLPs where CoT accelerates pretraining by learning composable shortcuts and filtering the relevant layer.

For this survey, the important point is that intermediate CoT states are not treated only as explanations of an already completed hidden computation. In this controlled setting, they change the computation available to the model and make composition more efficient. The result is therefore useful for understanding a possible computational role of intermediate tokens, but it should not be assumed to transfer directly to natural-language CoT in pretrained LLMs.

4.2 Reasoning in pretrained language models

Pretrained language models mix reasoning with stored knowledge, language regularities, redundant pathways, and prompt-dependent behavior. As a result, the studies below often explain an important part of the reasoning process rather than reconstructing one complete end-to-end algorithm.

4.2.1 Arithmetic and formal logic

These studies show different levels of detail in mechanistic explanations of pretrained models. In arithmetic, Stolfo et al. (2023) use causal mediation to trace how query-relevant quantities are routed toward the final position and how later MLPs contribute information related to the result. In propositional reasoning, Hong et al. (2025) connect several localized components into a broader information-flow account involving rule selection, routing, processing, and final decision support. Kim et al. (2025) analyze a narrower syllogistic mechanism in more depth, identifying a middle-term suppression circuit and testing necessity/sufficiency, transfer across syllogistic schemes, and interaction with belief bias. The comparison shows why finding a reasoning-related component and establishing a reusable causal mechanism should not be treated as equivalent claims.

4.2.2 Shared circuits and timing

Component identity alone does not fully characterize a reasoning mechanism. Two additional questions are whether a computation is reused across tasks and whether relevant information appears early enough to be used. Lan et al. (2024) find analogous subgraphs across related sequence-continuation tasks in GPT-2 Small and Llama-2 7B, providing evidence that some computations are reused across different surface forms. Biran et al. (2024) focus instead on timing in two-hop factual questions. In some failures, the correct bridge entity appears too late in depth for the remaining layers to complete the second retrieval; moving the representation earlier repairs some cases. Together, these results show that both circuit reuse and computational timing matter when explaining how a reasoning process succeeds.

4.3 What CoT reveals in pretrained models

Dutta et al. (2024) and Hou et al. (2023) approach a related question from different directions: how much of the reasoning process can be recovered from visible or internal intermediate structure? On multi-step reasoning over fictional ontologies, Dutta et al. (2024) find several parallel pathways that can contribute to the answer. They also observe a change across depth: earlier layers are more influenced by pretraining priors, while later layers rely more on the provided context. The visible CoT therefore changes the computational setting but does not provide a complete transcript of all hidden processing.

Hou et al. (2023) introduce MechanisticProbe, which tries to recover a reasoning tree from attention patterns on synthetic and language reasoning tasks. Their results suggest that procedural structure can be recovered from attention, but this is better viewed as representational evidence than as a full causal reconstruction. Read together, the two papers suggest that explicit reasoning can reveal useful computational organization without necessarily exposing the complete internal process that produces the answer.

4.4 Sparse features related to reasoning

The feature-level studies move from positive evidence for reasoning-related directions to a stricter question: whether the semantic interpretation assigned to those directions survives attempts to falsify it. Galichin et al. (2026) identify candidate reasoning-related features in DeepSeek-R1-family models and report that steering them changes reasoning behavior. Chen et al. (2026) find CoT-associated SAE features in Pythia and a scale-dependent causal effect from feature patching. These results show that feature-level directions can influence reasoning behavior, but they do not by themselves establish exactly what computation a feature represents.

Ma et al. (2026) directly test that stronger semantic claim. Lexical injections, counterexamples, paraphrases, and steering analyses show that apparently meaningful features can sometimes reflect discourse or lexical regularities rather than the reasoning operation assigned to them. The key disagreement is not whether the features are active or causally influential. It is whether that evidence is enough to justify calling them high-level reasoning features.

4.5 From findings to comparison

The studies reviewed above differ not only in the reasoning settings they examine, but also in how much of the underlying computation they explain. Section 5 compares these differences directly using the evidence categories from Section 3.2 and asks what is still missing before an important component becomes a fuller explanation of the reasoning process.

5 Comparative Analysis Across Studies

The previous section reviewed what individual studies found in different reasoning settings. This section compares those findings. The goal is not to find one mechanism shared by all reasoning tasks. Instead, we ask what kinds of explanations the papers support, what patterns appear more than once, and where those explanations remain incomplete.

Three questions guide the comparison. First, how far do the studies move from finding reasoning-related information to explaining the computation that uses it? Second, which computational patterns appear across several tasks, and where do those similarities break down? Third, can the proposed mechanisms help explain when reasoning succeeds, transfers, or fails? These questions connect the evidence categories in Section 3 with the empirical findings in Section 4.

5.1 From finding important components to explaining the full process

Table 2 summarizes the empirical reasoning studies using the evidence categories from Section 3.2. The table is meant to distinguish different kinds of support, not to rank papers. For example, decoding task-relevant information from a layer answers a different question from reconstructing how that information is turned into an answer or testing whether the proposed mechanism predicts failure on new inputs.

A broad difference appears across the reviewed settings. Controlled tasks more often allow reconstruction and explicit tests of where a mechanism stops working because the target computation can be specified and manipulated directly. Studies of pretrained language models more often provide localization, intervention, timing, or feature-level evidence without reconstructing the complete reasoning process. This difference reflects the difficulty of the setting rather than the quality of the studies. In the table, seven of eight controlled algorithmic studies include reconstruction and/or stress-testing evidence, compared with four of ten studies on pretrained or reasoning-tuned models; explicit Level 3 necessity/sufficiency evidence appears in only one tabled study. These numbers describe the papers reviewed here; they are not field-wide statistics or quality scores.

Table 2: Empirical reasoning studies. Evidence profile uses the levels in Section 3.2 non-cumulatively: 1=localization/decoding, 2=causal intervention, 3=necessity/sufficiency, 4=reconstruction, 5=stress testing/falsification.

Study	Setting	What is explained	Evidence	Main test or limit
Nanda et al. (2023)	Modular addition	Fourier circuit	1/2/4	Controlled algorithmic task
Wang et al. (2024)	Implicit composition/comparison	Memorizing and generalizing circuits	1/4/5	Generalization outside training differs by task
Todd et al. (2026)	In-context algebra	Copying/identity/cancellation	1/2/5	Strategy depends on prompt structure
Saparov et al. (2025)	Graph connectivity	Reachable-set updates	1/4/5	Larger graphs remain difficult
Zhang et al. (2025b)	CoT state tracking	Implicit FSA/state neurons	1/5	Length extrapolation
Cabannes et al. (2024)	Iterative CoT	Iteration head/update rule	1/4/5	Task-specific updates/alternative circuits
Li et al. (2023)	Compositional ICL	Filtering + single-step ICL	4	Controlled MLP setting
Maltoni & Ferrara (2026)	Synthetic deduction	Horizontal and vertical computation	1	Small controlled models
Stolfo et al. (2023)	Pretrained LM arithmetic	Routing + MLP processing	1/2	Arithmetic/model families tested
Hong et al. (2025)	Propositional reasoning	Locate–move–process–decide circuit	1/2/4	Formal tasks; organization varies by model
Kim et al. (2025)	Syllogistic inference	Middle-term suppression circuit	1/2/3/5	Belief bias affects formal mechanism
Lan et al. (2024)	Sequence continuation	Shared subcircuits	1/2/5	Related sequence families
Biran et al. (2024)	Two-hop factual reasoning	Bridge-entity timing	1/2	Depends on timing and model
Dutta et al. (2024)	Llama-2 7B CoT	Parallel pathways/depth roles	1	One model/benchmark setting
Hou et al. (2023)	Synthetic/language reasoning	Attention-based reasoning tree	1	Probe, not full causal reconstruction
Galichin et al. (2026)	Reasoning-tuned models	SAE reasoning-related features	1/2	Feature meaning depends on validation
Chen et al. (2026)	Pythia CoT/no-CoT	CoT-associated SAE features	1/2	Depends strongly on scale and task
Ma et al. (2026)	SAE candidates	Candidate reasoning features	1/2/5	Tests whether feature interpretations survive controls

Figure 1 shows the same comparison from another angle by counting how often each evidence type appears in the two settings. Because the categories are non-cumulative, one study can contribute to several bars; the counts therefore describe evidence profiles rather than mutually exclusive levels.

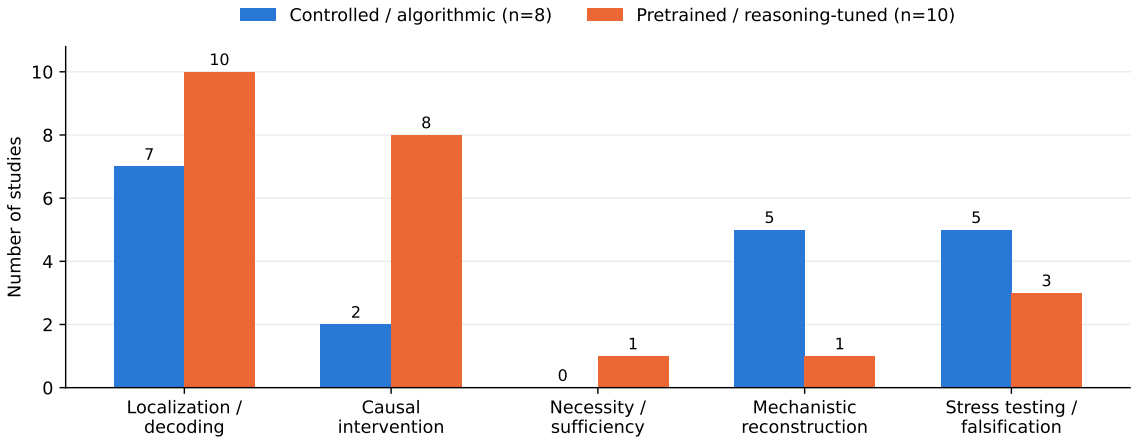

Figure 1: Evidence types across the empirical studies in Table 2. Bars show the number of studies containing each type of mechanistic evidence. Categories are non-cumulative, so a study may contribute to multiple bars. Controlled/algorithmic studies: $n = 8$; pretrained/reasoning-tuned studies: $n = 10$.

The table makes the main difference easy to see: controlled studies more often reconstruct an algorithm or test a clear limit, while several pretrained-model studies isolate a narrower component, pathway, timing

effect, or feature. There are important exceptions, such as syllogistic necessity/sufficiency tests and SAE falsification. The pattern should therefore be read as a broad comparison, not a fixed rule.

5.2 Common computational patterns and where they break down

Several studies can be compared using the vocabulary of retrieval, routing, state updating, and decision-making. Arithmetic work separates routing from later result processing (Stolfo et al., 2023); propositional logic contains locate/move/process/decide components (Hong et al., 2025); iteration heads separate routing from state update (Cabannes et al., 2024); compositional ICL separates filtering from a step-level learning operation (Li et al., 2023); and two-hop factual reasoning requires a bridge entity before the second retrieval (Biran et al., 2024). Related sequence, logic, iteration, and syllogistic tasks also show some reuse of subcircuits (Lan et al., 2024; Hong et al., 2025; Cabannes et al., 2024; Kim et al., 2025).

These terms are useful for comparison, but they do not fit every mechanism. Fourier modular addition is better described as a distributed transformation (Nanda et al., 2023); grokked comparison and composition develop different circuit organizations (Wang et al., 2024); graph reachability uses iterative updates without four clean modules (Saparov et al., 2025); and in-context algebra can switch among strategies (Todd et al., 2026). The safer conclusion is that some operations are reused in task-dependent ways, not that all reasoning follows one general circuit.

5.3 Can mechanisms predict timing, transfer, and failure?

A mechanistic explanation is more useful when it helps predict not only *what* computation happens, but also *when* and *where* it may fail. Biran et al. (2024) show that a correct bridge entity may become available too late in depth for a second retrieval; Dutta et al. (2024) find depth-dependent roles across CoT; and SICAF tracks token influence through depth (Zhang et al., 2025a). These results show why a static statement such as "layer L contains concept X" can miss the way information develops over time.

The same issue appears when inputs move outside the original setting. Graph-search performance degrades on larger graphs despite an interpretable search-like algorithm (Saparov et al., 2025); FSA state tracking retains a length limit (Zhang et al., 2025b); grokked comparison and composition behave differently out of distribution (Wang et al., 2024); while in-context algebra transfers to unseen groups under a carefully designed setup (Todd et al., 2026). The strongest explanations do more than explain successful examples after the fact: they also help predict where the mechanism will continue to work and where it will break.

5.4 What does CoT tell us about computation?

Controlled work shows that intermediate states can add computation. Li et al. (2023) distinguish the benefit of seeing intermediate states from recurrently generating and reusing them. Iteration and FSA studies likewise use explicit states as part of repeated computation (Cabannes et al., 2024; Zhang et al., 2025b), while Maltoni & Ferrara (2026) distinguish token-based horizontal reasoning from depth-based vertical reasoning. In pretrained models, CoT changes information flow but does not provide a complete transcript of hidden pathways (Dutta et al., 2024). Taken together, these studies suggest that CoT can be part of the computation, while the question of whether the visible text faithfully describes hidden computation remains separate.

5.5 How reliable are sparse reasoning features?

The SAE studies show one of the clearest disagreements in the reviewed literature. Candidate features can align with reasoning-related behavior and respond to steering or patching (Galichin et al., 2026; Chen et al., 2026). However, lexical injections, counterexamples, and paraphrases can break interpretations that initially look meaningful (Ma et al., 2026). A feature may represent a reasoning operation, a textual marker, or a higher-level control signal, and steering can affect all three. Strong claims about feature meaning therefore need semantic controls and attempts to falsify the interpretation, not only intuitive labels or downstream effects.

5.6 Why controlled tasks allow deeper explanations

The comparison above also helps explain why the evidence profiles differ by setting. In controlled tasks, researchers usually know the task rules, relevant state variables, or target algorithm. This makes it easier to define intermediate states, construct targeted counterexamples, and test whether a proposed mechanism reproduces the computation. In pretrained models, the target computation is rarely known in advance, and several internal pathways may support the same behavior. As a result, localization or intervention results in pretrained models often answer a narrower but more difficult question than full reconstruction in a controlled task.

6 Open Questions and Future Directions

The questions below come directly from limits or disagreements in the reviewed papers. For each question, we state what is missing now, what stronger evidence would show, and one possible test. To indicate how broadly each issue appears in this survey, we use four labels: **narrow** for one or two focal studies, **several studies** when at least three studies or task settings show a similar issue, **broad but indirect** when many papers make the question relevant but few test it directly, and **concentrated conflict** for a small set of closely related papers with results that need direct comparison.

6.1 Do interventions and CoT reflect the model's usual computation?

Support in reviewed papers: several studies — related concerns appear in both intervention studies and CoT studies.

What is missing now. Activation patching can restore behavior even when the intervention pushes the model into a state it would not naturally reach (Heimersheim & Nanda, 2024; Grant et al., 2026). CoT creates a related ambiguity. Controlled studies show that generated intermediate states can genuinely add computation (Li et al., 2023; Cabannes et al., 2024; Zhang et al., 2025b), but analyses of pretrained models do not show that every generated step is a direct readout of the hidden pathway that produced the answer (Dutta et al., 2024).

What stronger evidence would show. An intervention should not only repair the final answer; it should also restore a downstream state trajectory similar to natural successful runs. Likewise, a claimed CoT state should remain tied to the same hidden computational state when the wording changes, and should change when the underlying state changes.

One possible test. Use a task with a known latent state and compare three conditions: the state remains hidden, the state is shown, or the model generates and reuses it. Apply several intervention schemes and measure both final behavior and downstream hidden states. This would help separate recovery of the model's usual computation from success created by the intervention itself.

6.2 Can explanations at different levels agree?

Support in reviewed papers: broad but indirect — many papers explain reasoning at different levels, but few compare those levels on the same task.

What is missing now. Arithmetic and logic studies often focus on attention heads and MLPs (Stolfo et al., 2023; Hong et al., 2025); automated methods identify edge-level circuits (Conmy et al., 2023; Hsu et al., 2025; Wu & Barez, 2026); SAE work uses sparse features and feature circuits (Marks et al., 2025; Chen et al., 2026); and controlled studies sometimes recover higher-level objects such as an FSA or iteration rule (Zhang et al., 2025b; Cabannes et al., 2024). It is often unclear whether these are compatible views of the same computation or only partly overlapping explanations.

What stronger evidence would show. Explanations at different levels should make consistent predictions. A feature should move through the circuit edges claimed to carry it; changing a component should alter the higher-level state in the expected way; and the combined low-level account should reproduce the algorithmic behavior on new inputs.

One possible test. Start with a task that already has a well-supported component-level circuit. Learn SAE features on the relevant activations, track those features through the proposed edges, and intervene at both feature and component levels. Then compare the resulting internal state with the state predicted by the higher-level algorithm. Agreement would support the idea that the different explanations describe the same computation at different levels.

6.3 Can mechanisms predict when reasoning will transfer or fail?

Support in reviewed papers: several studies — transfer, strategy changes, OOD limits, and timing failures appear across several task families.

What is missing now. Some papers find reusable subcircuits across related tasks (Lan et al., 2024; Hong et al., 2025; Cabannes et al., 2024; Kim et al., 2025), while others find strategy switching or clear OOD limits (Todd et al., 2026; Wang et al., 2024; Saparov et al., 2025; Zhang et al., 2025b). Timing can create a similar problem within a single example: the right intermediate information may appear, but too late for the remaining layers or tokens to finish the computation (Biran et al., 2024).

What stronger evidence would show. A mechanistic explanation should help predict new behavior before the final answer is known. It should imply a limit based on available depth, sequence length, state capacity, or variable-binding structure, and then correctly predict where the model stops generalizing.

One possible test. Identify a mechanism in one setting and use it to predict a limit before evaluating new data. Then test held-out examples on both sides of that limit while tracking an intermediate progress variable, such as bridge-entity availability, FSA state, or reachable-set maturity. A useful mechanism should predict both broad distribution limits and which individual examples are likely to fail.

6.4 How can we test whether a reasoning feature really means what we think it means?

Support in reviewed papers: concentrated conflict — a small set of feature papers supports different interpretations under different validation tests.

What is missing now. Galichin et al. (2026) and Chen et al. (2026) show that candidate sparse features can influence reasoning-related behavior through steering or patching. Ma et al. (2026) show that apparently meaningful interpretations can fail under lexical injections, counterexamples, and paraphrases. Steering alone therefore cannot tell us whether a feature represents a reasoning operation, a discourse marker, or a higher-level control signal.

What stronger evidence would show. A reasoning feature should keep the same interpretation when wording changes but meaning does not. It should distinguish reasoning from carefully matched non-reasoning text, respond correctly to semantic counterexamples, and still produce the expected causal effect when placed in a larger circuit-level explanation.

One possible test. Discover candidate features on one family of reasoning traces and fix the feature labels before further evaluation. Test them on lexical injections, paraphrases, matched controls, semantic counterexamples, and a second reasoning task. Only features that survive those tests should then be used for steering or circuit integration.

7 Limitations of This Survey

This is a focused rather than exhaustive review. Adjacent work on CoT faithfulness, general induction/copying, other interpretability methods, non-transformer architectures, and closed frontier-scale systems is not covered systematically. Model coverage is also uneven: the reviewed pretrained studies range from GPT-2 and Pythia to Llama-family and reasoning-tuned open models, while the clearest end-to-end reconstructions often come from small controlled transformers. The reviewed evidence therefore does not establish that mechanisms found in these settings remain unchanged at frontier scale.

The methods are not evaluated under one common benchmark. Table 1 explains what ACDC, CD-T, Query Circuits, and SICAF measure, but it should not be read as a ranking of reliability or efficiency. Likewise, the evidence profiles in Table 2 are our way of comparing the reviewed papers; they are not standardized scores or a meta-analysis.

Finally, many reviewed reasoning tasks are intentionally simplified. This supports cleaner causal analysis but limits direct transfer to open-ended reasoning that depends on world knowledge, long contexts, or more complex post-training. This survey adds no new experiments. Its contribution is comparative: it organizes the reported evidence, separates stronger from narrower mechanistic claims, and derives future questions from concrete disagreements and tested limits in the reviewed studies.

8 Conclusion

The reviewed papers show that transformer reasoning does not have one simple mechanistic story. In controlled settings, researchers can sometimes recover clear algorithms, such as Fourier-style modular addition, layer-by-layer graph reachability, finite-state tracking, and iterative update mechanisms. In pretrained language models, reasoning is usually more distributed and harder to reconstruct. Generated intermediate steps can also change both the information available to the model and the computation it can perform (Li et al., 2023; Dutta et al., 2024).

The main lesson of this survey is that mechanistic findings differ in how much they explain. Among the empirical studies in Table 2, seven of eight controlled studies include reconstruction or stress-testing evidence, compared with four of ten studies on pretrained or reasoning-tuned models; explicit necessity/sufficiency evidence appears in only one tabled study. These numbers describe the papers reviewed here rather than the wider field, but they make the contrast clear.

Finding an interpretable component is not the same as explaining the full computation. Stronger explanations connect components into a causal process, survive attempts to challenge the interpretation, and help predict where the model will transfer or fail. Ideas such as retrieval, routing, and state updating are useful for comparing some papers, but the reviewed studies also show distributed computation, strategy switching, and task-specific circuit organization. Progress in mechanistic interpretability of reasoning therefore depends not only on finding more internal structure, but on building explanations that remain useful under stronger causal, transfer, and failure tests.

References

Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. Hopping too late: Exploring the limitations of large language models on multi-hop queries. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14113–14130. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.781. URL https://aclanthology.org/2024.emnlp-main.781/.

Vivien Cabannes, Charles Arnal, Wassim Bouaziz, Alice Yang, Francois Charton, and Julia Kempe. Iteration head: A mechanistic study of chain-of-thought. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/c50f8180ef34060ec59b75d6e1220f7a-Abstract-Conference.html.

Xi Chen, Aske Plaat, and Niki van Stein. How does chain of thought think? mechanistic interpretability of chain-of-thought reasoning with sparse autoencoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30297–30305, 2026. doi: 10.1609/aaai.v40i36.40281. URL https://ojs.aaai.org/index.php/AAAI/article/view/40281.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adria Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Abstract-Conference.html.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

Subhabrata Dutta, Joykirat Singh, Soumen Chakrabarti, and Tanmoy Chakraborty. How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=uHLDkQVtyC.

Andrey V. Galichin, Alexey Dontsov, Polina Druzhinina, Anton Razzhigaev, Oleg Y. Rogov, Elena Tutubalina, and Ivan V. Oseledets. I have covered all the bases here: Interpreting reasoning features in large language models via sparse autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30771–30779, 2026. doi: 10.1609/aaai.v40i36.40334. URL https://ojs.aaai.org/index.php/AAAI/article/view/40334.

Satchel Grant, Simon Jerome Han, Alexa R. Tartaglini, and Christopher Potts. Addressing divergent representations from causal interventions on neural networks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=cZrTMqYVL6.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024. URL https://arxiv.org/abs/2404.15255.

Guan Zhe Hong, Nishanth Dikkala, Enming Luo, Cyrus Rashtchian, Xin Wang, and Rina Panigrahy. A implies b: Circuit analysis in llms for propositional logical reasoning. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=M0U8wUow8c.

Yifan Hou, Jiaoda Li, Yu Fei, Alessandro Stolfo, Wangchunshu Zhou, Guangtao Zeng, Antoine Bosselut, and Mrinmaya Sachan. Towards a mechanistic interpretation of multi-step reasoning capabilities of language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4902–4919. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.299. URL https://aclanthology.org/2023.emnlp-main.299/.

Aliyah R. Hsu, Georgia Zhou, Yeshwanth Cherapanamjeri, Yaxuan Huang, Anobel Y. Odisho, Peter R. Carroll, and Bin Yu. Efficient automated circuit discovery in transformers using contextual decomposition. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=41HlN8XYM5.

Geonhee Kim, Marco Valentino, and Andre Freitas. Reasoning circuits in language models: A mechanistic interpretation of syllogistic inference. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10074–10095. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-acl.525/.

Michael Lan, Philip Torr, and Fazl Barez. Towards interpretable sequence continuation: Analyzing shared circuits in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12576–12601. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.699. URL https://aclanthology.org/2024.emnlp-main.699/.

Yingcong Li, Kartik Sreenivasan, Angeliki Giannou, Dimitris Papailiopoulos, and Samet Oymak. Dissecting chain-of-thought: Compositionality through in-context filtering and learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 22021–22046, 2023. doi:

10.52202/075280-0966. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/45e15bae91a6f213d45e203b8a29be48-Abstract-Conference.html.

George Ma, Zhongyuan Liang, Irene Y. Chen, and Somayeh Sojoudi. Do sparse autoencoders identify reasoning features in language models? In *Forty-Third International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2601.05679.

Davide Maltoni and Matteo Ferrara. Deductive logic in language models: Horizontal vs. vertical reasoning. *Machine Learning and Knowledge Extraction*, 8(7):214, 2026. doi: 10.3390/make8070214. URL https://www.mdpi.com/2504-4990/8/7/214.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=I4e82CIDxv.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022. URL https://arxiv.org/abs/2201.02177.

Abulhair Saparov, Srushti Pawar, Shreyas Pimpalgaonkar, Nitish Joshi, Richard Yuanzhe Pang, Vishakh Padmakumar, Seyed Mehran Kazemi, Najoung Kim, and He He. Transformers struggle to learn to search. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9cQB1Hwrtw.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7035–7052. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.435. URL https://aclanthology.org/2023.emnlp-main.435/.

Eric Todd, Jannik Brinkmann, Rohit Gandikota, and David Bau. In-context algebra. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=J2peqXPQbB.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokked transformers are implicit reasoners: A mechanistic journey to the edge of generalization. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=ns8IH5Sn5y.

Tung-Yu Wu and Fazl Barez. Query circuits: Explaining how language models answer user prompts. In *Forty-Third International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2509.24808.

Lin Zhang, Lijie Hu, and Di Wang. Mechanistic unveiling of transformer circuits: Self-influence as a key to model reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1387–1404. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.findings-naacl.76. URL https://aclanthology.org/2025.findings-naacl.76/.

Yifan Zhang, Wenyu Du, Dongming Jin, Jie Fu, and Zhi Jin. Finite state automata inside transformers with chain-of-thought: A mechanistic study on state tracking. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13603–13621. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.acl-long.668. URL https://aclanthology.org/2025.acl-long.668/.