

Mechanistic Interpretability of In-Context Learning:

A Survey of Known Circuits and Open Problems

Authors

Lead Researcher: Pranay M Mahendrakar

Mail: pranaymahendrakar2001@gmail.com

Contact: +91 6361723454

Company: SONYTECH

Date: April 2026

Abstract

In-context learning (ICL) — the ability of a transformer language model to acquire a new input-output mapping from a handful of demonstrations in its prompt, with no weight updates — remains the most striking and least understood capability of large language models. Mechanistic interpretability has produced two influential partial explanations: induction-head circuits, which implement fuzzy pattern completion of the form $[A][B] \dots [A] \rightarrow [B]$, and function vectors, which compress an entire ICL task into a low-dimensional representation routed by a small number of attention heads. A third line of work argues that ICL implements implicit gradient descent on linear regression problems. This paper contributes (i) a structured synthesis of these three accounts, (ii) a sharper articulation of where each explanation succeeds and where it breaks down, and (iii) a concrete research agenda targeting five gaps that current work does not address: compositional ICL, ICL in instruction-tuned frontier models, the transient-versus-persistent learning distinction, the polysemanticity barrier at scale, and the deeper interpretive question of what it means to claim that a circuit "explains" a behaviour. We argue the field has solved the easy cases and now faces the hard ones.

Keywords

Mechanistic interpretability, in-context learning, induction heads, function vectors, transformer circuits, attention mechanisms, language models, AI safety

1. Introduction

Few empirical phenomena in modern machine learning have been as surprising as in-context learning (ICL). A transformer language model, trained only on next-token prediction, can be shown three or four examples of a task it has never explicitly seen — say, translating English words into a fictional cipher — and it will then complete the next example correctly, often at near task-specific-fine-tuning quality. The model's weights do not change. The learning, whatever it is, happens entirely in the forward pass.

This is not a marginal capability. It is the core mechanism behind modern prompting, agentic workflows, tool use, and most practical deployment of frontier models. And yet, more than five years after Brown et al. (2020) named the phenomenon, the field still cannot give a complete circuit-level account of how it works in a frontier-scale model. We have explanations for fragments. We do not have an explanation for the whole.

This paper makes three contributions. First, it synthesizes the three most developed mechanistic accounts of ICL —

induction heads, function vectors, and the implicit-gradient-descent hypothesis — and shows how they relate. These are often discussed in isolation; in fact they describe complementary and partially overlapping phenomena, and clarifying the relationship is itself useful work. Second, it sharpens the question of what remains unexplained. Much of the existing literature concedes that ICL is "not yet fully understood" without specifying what would constitute fuller understanding. We identify five concrete gaps. Third, it proposes a research agenda — specific experiments, methodological commitments, and one harder theoretical question — that takes those gaps seriously rather than restating them.

A note on scope. This paper is a survey and position paper; it does not present new empirical results. The contribution is conceptual: a clearer map of the territory and a more precise statement of the problems that remain. We believe such maps are valuable in a field that has, in recent years, accumulated empirical findings faster than it has organized them.

2. Background

2.1 In-Context Learning: A Working Definition

We adopt the following operational definition. A transformer model M exhibits in-context learning on a task T if, given a prompt consisting of k demonstrations of T followed by a held-out query, its accuracy on the query is meaningfully higher than its zero-shot accuracy on the same query without demonstrations. The strength of ICL on T is the magnitude of this gap as a function of k .

Three features of this definition deserve emphasis. First, ICL is defined behaviourally, not mechanistically — a model could in principle achieve the behavioural signature through many different internal mechanisms, and disentangling them is precisely the interpretability problem. Second, ICL is task-relative; a model may show strong ICL on one task family and none on another, which itself is informative. Third, the learning is transient: nothing in the model's weights records that it ever encountered T . The next prompt starts from a blank slate.

It is this last feature that makes ICL genuinely puzzling from a classical machine-learning perspective. There is no training set, no loss, no gradient step. Whatever computation implements the adaptation must be implementable purely as activations flowing through a fixed network.

2.2 Mechanistic Interpretability: Tools and Goals

Mechanistic interpretability (MI) is the project of reverse-engineering the algorithms a neural network has learned to implement, at the level of weights, neurons, and circuits. The methods include residual-stream decomposition, attention-pattern analysis, activation patching and causal mediation, sparse autoencoders for feature decomposition, and increasingly, automated circuit discovery.

The standard of evidence in MI has tightened considerably. Early work was satisfied with finding components whose ablation degraded a behaviour; current work demands causal interventions that demonstrate sufficiency as well as necessity, ideally accompanied by a precise mathematical description of what the circuit computes. This raises the bar — and, as we will argue in Section 4, it raises it in ways that frontier-scale models do not yet meet.

3. Known Mechanisms of In-Context Learning

3.1 Induction Heads

The most influential mechanistic account of ICL comes from Olsson et al. (2022), who identified a two-layer attention circuit they named the induction head. In its simplest form, an induction head implements the pattern $[A][B] \dots [A] \rightarrow [B]$:

when the model encounters a token A , the circuit attends to the previous occurrence of A elsewhere in the context, looks at what token followed it (B), and copies B as the prediction for the current position.

The circuit decomposes cleanly. A previous-token head in an early layer attends to the immediately preceding token and writes a representation of it into the current position's residual stream. A later induction head uses this signal: its query attends to positions whose key encodes "the token before me was A ," and its OV circuit copies the value of the matched position into the output. The two heads compose to implement match-and-copy.

Olsson et al. presented six lines of evidence that induction heads are the dominant mechanism behind ICL in small transformers. The most striking is a phase change in training: induction heads form abruptly at a specific point during pretraining, and ICL ability — measured as the loss difference between early and late tokens in a sequence — improves sharply at the same moment. This co-occurrence is highly suggestive of a causal relationship, and ablation experiments in small attention-only models confirm it.

Subsequent work has both broadened and qualified this picture. Singh et al. (2024) showed that the formation of induction circuits depends sensitively on the structure of the training data and on the interaction between attention heads. Reddy (2024) provided a more detailed account of why the phase transition is sharp. Crucially, however, the original paper itself was careful to note that the strong causal evidence applied to small attention-only models; for larger models with MLPs, the evidence was correlational. This caveat is often forgotten in subsequent citations and is one of the gaps we return to in Section 4.

3.2 Function Vectors and Task Representations

A different and partially complementary account emerged from work by Todd et al. (2024) and, independently and concurrently, Hendel et al. (2023). Both groups discovered that an ICL prompt induces a compact vector representation of the task in the model's hidden states — a function vector or task vector — that can be extracted from a specific layer and then transplanted into a different context to trigger the same task behaviour, even without any demonstrations in the new context.

Todd et al. used causal mediation analysis to identify a small set of attention heads (typically in middle layers) that are responsible for transporting these vectors. Hendel et al. showed that the existence of such vectors is consistent with treating ICL as a two-step process: the model first compresses the demonstrations into a task representation $\theta(S)$, then applies a function $f(x; \theta(S))$ to the query.

The function-vector account is not in conflict with the induction-head account; it operates at a different level of description. Induction heads describe a low-level attentional mechanism (match-and-copy). Function vectors describe a higher-level representational format (a compressed task encoding). It is plausible — though, importantly, not yet proven — that induction-like circuits in early layers contribute to forming the function vector, which is then deployed by FV heads in middle layers. Reconciling the two accounts at a mechanistic level remains open work.

3.3 The Implicit Gradient Descent Hypothesis

A third account, developed in parallel by Akyürek et al. (2023), von Oswald et al. (2023), and Dai et al. (2023), takes a different angle. These papers argue that, on certain restricted tasks (notably linear regression with prompts of the form $(x_1, y_1), (x_2, y_2), \dots, x_{\text{query}}$), a single self-attention layer is mathematically equivalent to one step of gradient descent on a least-squares loss. Stacking attention layers therefore corresponds to running multiple gradient steps. On this view, the transformer is, internally and implicitly, executing the standard learning algorithm on the demonstrations — just compiled into the forward pass.

This is a beautiful result and it almost certainly captures something real, but its scope is narrower than initial enthusiasm suggested. The equivalence holds cleanly for linear self-attention on linear regression. For softmax attention on natural-language tasks, the relationship is much looser; the model is not literally doing gradient descent, only doing something whose mathematical structure resembles it under simplifying assumptions. Subsequent work has debated how far the analogy extends.

3.4 How These Accounts Relate

It is tempting to ask which account is correct, but this is the wrong question. Each account is doing something different. Induction heads describe one specific circuit pattern that demonstrably participates in ICL. Function vectors describe a representational format that ICL produces. The gradient-descent hypothesis describes an algorithmic abstraction that some attention configurations implement on some tasks. None of them, individually or collectively, gives a full account of how a frontier model does ICL on, say, a multi-step word problem with novel premises.

The honest summary is this: we have a strong mechanistic story for the simplest form of ICL — copying a demonstrated pattern — and progressively weaker stories for everything else. This is the gap the next section addresses.

4. Open Problems

4.1 Compositional and Multi-Step ICL

Induction heads explain how a model can extend a demonstrated pattern. They do not explain how a model can compose two demonstrated patterns into a third it has never seen, or how it can apply a demonstrated rule to a novel input that requires multi-step reasoning. Yet frontier models routinely do both.

Consider a prompt that demonstrates a custom arithmetic operation on three examples and then asks the model to apply it to a new pair, where solving the new pair requires recognizing that the operation distributes over a familiar one. No published induction-head or function-vector circuit accounts for this. Either the existing circuits compose in ways we have not characterized, or there are additional circuits whose discovery has not yet been made. Both possibilities are interesting; neither has been ruled out.

4.2 ICL in Instruction-Tuned and RLHF'd Frontier Models

Almost all detailed circuit-level work on ICL has been done on small or medium base models — pretrained-only transformers between roughly 1M and 7B parameters. The frontier deployed models that users actually interact with have been instruction-tuned, RLHF'd, and often subjected to additional alignment training. Whether the same circuits exist in those models, whether they are augmented by new circuits introduced by post-training, and whether post-training changes the relative importance of induction-style versus function-vector-style mechanisms — none of these has been established.

This matters for safety as well as for science. Claims about model behaviour grounded in interpretability evidence on base models do not automatically transfer. A safety property that holds because of a known circuit in a base model may or may not hold after RLHF. Closing this gap requires either scaling existing methods to frontier models — currently expensive and partially blocked by polysemanticity, discussed below — or developing methods that can recover circuits with weaker assumptions about model size.

4.3 The Transient Versus Persistent Learning Distinction

ICL is, by definition, transient: it leaves no trace in the weights. But this raises a question that the existing literature largely sidesteps. When a model "learns" a new task in-context, is it constructing a genuinely novel computational structure in its activations, or is it merely retrieving and parameterizing a structure that was already latent in its

weights? The function-vector account leans toward the latter: the demonstrations select a vector from a space the model already represents. The induction-head account is silent on the question, and so is the gradient-descent account.

The distinction has real consequences. If ICL is essentially retrieval of pre-formed task representations, then the space of tasks a model can learn in-context is bounded by what was implicit in its training. If ICL is genuine in-activation computation, the space is potentially much larger. We do not know which view is closer to the truth, and we do not have a clean experimental design to settle it. Designing one is an obvious target for future work.

4.4 The Polysemanticity Barrier

Mechanistic interpretability at scale is currently bottlenecked by superposition: individual neurons and even individual directions in the residual stream represent multiple unrelated features. This means that when we identify a "circuit" in a large model, we may be identifying an entanglement of multiple circuits whose interactions we have not disambiguated.

Sparse autoencoders have made significant recent progress on this problem, decomposing model activations into sparser, more monosemantic features. But the project of building circuit-level accounts on top of SAE features is still in early days, and applying it specifically to the components implicated in ICL has barely begun. Until this matures, claims of the form "circuit C implements behaviour B in model M" should be treated as provisional whenever M is at frontier scale.

4.5 The Interpretive Question: What Does a Circuit Explain?

This is the deepest open problem and the one the field is most reluctant to confront directly. Suppose we identify a circuit C in a model and demonstrate that ablating C destroys behaviour B and that activating C (in a clean prompt) elicits B. Have we explained how the model does B?

There is a sense in which we have. We have located the components, characterized the algorithm, and verified causality. But there is another sense in which we have only pushed the question back. Why does activating those particular weights produce that algorithm? What in the training process made those weights take those values? The mechanistic account terminates at the level of the trained network; it does not by itself tell us about the inductive biases, data statistics, or optimization dynamics that produced the network.

This is not an objection to mechanistic interpretability — every level of explanation has to stop somewhere. It is a flag that the term "explanation" is being used in a more restricted sense than non-specialists may assume, and that articulating that restriction matters when we make safety claims grounded in MI evidence.

5. A Research Agenda

5.1 Methodological Recommendations

Several methodological commitments would, in our view, accelerate progress. First, papers claiming to identify ICL circuits should default to causal interventions (activation patching, ablation, transplantation) rather than correlational evidence, and should clearly distinguish between sufficiency and necessity claims. Second, results should be reported on at least one frontier-scale instruction-tuned model in addition to whatever small base model the analysis was developed on; where this is infeasible, the limitation should be stated explicitly rather than minimized. Third, the relationship between newly identified circuits and the existing induction-head and function-vector literature should be addressed directly — not by citing them, but by either subsuming them, distinguishing from them, or showing how they compose.

5.2 Specific Empirical Questions

Five questions seem tractable enough to attack in the near term:

1. Do induction heads and function-vector heads share components, or are they distinct subnetworks? A direct circuit-overlap analysis on a single model would settle this.
2. How does instruction-tuning modify ICL circuits? Comparing the same model before and after instruction-tuning, on the same ICL benchmark, with the same MI methods, should be feasible and would be highly informative.
3. Can a frontier model perform ICL on a task whose demonstrations require composition of two sub-tasks the model has never seen composed? If yes, what does the circuit look like? If no, where does it fail?
4. To what extent is in-context learning retrieval of pre-formed task vectors versus in-activation construction of new ones? A controlled experiment using carefully constructed novel tasks and probing the function-vector space could discriminate these hypotheses.
5. Do post-training procedures (instruction tuning, RLHF, constitutional AI) introduce new ICL circuits, modify existing ones, or leave them unchanged?

5.3 The Harder Theoretical Question

Beyond the empirical agenda, there is a theoretical question worth stating clearly: what is the right notion of "learning" for a system that learns within a single forward pass? Classical learning theory was built around the assumption that learning involves a change of internal state that persists. ICL violates this assumption. Either we extend classical theory to accommodate the new case, or we develop a separate theory for transient in-context computation, or we argue that what ICL does should not be called learning at all and find a better name. None of the three options is currently developed enough to evaluate. This is an opportunity.

6. Conclusion

The mechanistic interpretability of in-context learning is in an interesting position: it has produced several beautiful partial results, each of which captures something real, and it has not yet produced — and may not be close to producing — a complete account. The induction-head, function-vector, and implicit-gradient-descent accounts together explain perhaps the easiest 30% of the phenomenon. What remains is harder.

This is not a discouraging conclusion. It is an accurate one, and accuracy about where the open problems lie is the precondition for working on them productively. The goal of this paper has been to articulate those problems precisely enough that they can be attacked. The next phase of the field's work, we suggest, lies less in finding additional circuits in toy settings and more in three things: scaling the methods to frontier models, characterizing the circuits introduced by post-training, and developing a sharper theoretical framework for what it even means to learn within a single forward pass.

None of this will be quick. None of it is impossible.

References

- Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., & Zhou, D. (2023). What learning algorithm is in-context learning? Investigations with linear models. *International Conference on Learning Representations (ICLR)*.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 1877–1901.
- Conmy, A., Mavor-Parker, A., Lynch, A., Heimersheim, S., & Garriga-Alonso, A. (2023). Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems (NeurIPS)*, 36.
- Dai, D., Sun, Y., Dong, L., Hao, Y., Sui, Z., & Wei, F. (2023). Why can GPT learn in-context? Language models implicitly perform gradient descent as meta-optimizers. *Findings of the Association for Computational Linguistics: ACL 2023*, 4005–4019.
- Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., et al. (2021). A mathematical framework for transformer circuits. *Anthropic Transformer Circuits Thread*.
- Garg, S., Tsipras, D., Liang, P., & Valiant, G. (2022). What can transformers learn in-context? A case study of simple function classes. *Advances in Neural Information Processing Systems (NeurIPS)*, 35.
- Hendel, R., Geva, M., & Globerson, A. (2023). In-context learning creates task vectors. *Findings of the Association for Computational Linguistics: EMNLP 2023*, 9318–9333.
- Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., et al. (2022). In-context learning and induction heads. *Anthropic Transformer Circuits Thread*. arXiv:2209.11895.
- Reddy, G. (2024). The mechanistic basis of data dependence and abrupt learning in an in-context classification task. *International Conference on Learning Representations (ICLR)*.
- Singh, A. K., Moskovitz, T., Hill, F., Chan, S. C. Y., & Saxe, A. M. (2024). What needs to go right for an induction head? A mechanistic study of in-context learning circuits and their formation. *International Conference on Machine Learning (ICML)*.
- Todd, E., Li, M. L., Sharma, A. S., Mueller, A., Wallace, B. C., & Bau, D. (2024). Function vectors in large language models. *International Conference on Learning Representations (ICLR)*.
- von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., & Vladymyrov, M. (2023). Transformers learn in-context by gradient descent. *International Conference on Machine Learning (ICML)*.
- Wang, K., Variengien, A., Conmy, A., Shlegeris, B., & Steinhardt, J. (2023). Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. *International Conference on Learning Representations (ICLR)*.