Dissecting In-Context Learning: A Mechanistic Analysis of Emergent Circuits in Small Language Models

Anonymous authors
Paper under double-blind review

Abstract

In-context learning (ICL) enables language models to adapt to new tasks from just a few examples, yet the mechanistic basis of this capability remains poorly understood. We present a comprehensive analysis of the circuits underlying ICL in transformer models ranging from 125M to 1.3B parameters. Through systematic interventions and causal analysis, we identify four distinct circuit types that emerge during training: *copy circuits* that replicate patterns, *induction circuits* that abstract rules, *composition circuits* that combine information, and *task recognition circuits* that identify problem types. We demonstrate that these circuits are (1) causally responsible for ICL performance through targeted ablations showing 73% average performance degradation, (2) transferable across model scales with 0.82 correlation in circuit structure, and (3) surgically enhanceable, achieving 28% improvement on targeted tasks. Our analysis reveals that ICL emerges through the coordinated interaction of 12–15 critical attention heads forming interpretable computational graphs. We provide an open-source toolkit for ICL circuit analysis and demonstrate applications to model debugging and capability enhancement. These findings offer actionable insights for improving model interpretability and engineering more capable systems.

1 Introduction

Large language models (LLMs) demonstrate a remarkable ability to learn new tasks from just a few examples provided in their input context, a phenomenon known as in-context learning (Brown et al., 2020). This capability, which emerges without any parameter updates, fundamentally challenges our understanding of how neural networks process and generalize from information. Despite its practical importance and theoretical implications, the mechanistic basis of ICL—the specific computational circuits that enable it—remains largely opaque.

Consider a model shown examples like "cat → animal, dog → animal, rose → plant" and then correctly inferring "daisy → plant". This requires multiple sophisticated operations: pattern recognition, abstraction of the mapping rule, and application to novel inputs. How do transformer architectures, trained only to predict next tokens, develop such structured reasoning capabilities? Previous work has identified individual components like induction heads (Olsson et al., 2022) but lacks a comprehensive understanding of how these components interact to enable ICL.

In this work, we present a systematic mechanistic analysis of ICL in transformer language models. Our approach combines three key methodological innovations:

1. **Systematic Circuit Discovery**: We develop automated methods to identify and categorize attention head patterns associated with ICL across 47 different task types, revealing consistent circuit motifs.

2. **Causal Validation**: Through targeted interventions including activation patching, attention knockout, and circuit transplantation, we establish causal relationships between identified circuits and ICL performance.
3. **Circuit Manipulation**: We demonstrate that understanding these mechanisms enables practical applications, including debugging failure modes and enhancing model capabilities on specific tasks.

Our analysis reveals that ICL is implemented through the coordinated action of four distinct circuit types, each serving a specific computational role. These circuits emerge consistently across model scales and task domains, suggesting fundamental architectural principles. We find that only 12–15 attention heads (less than 8% of total heads in a 12-layer model) are critical for ICL, with their removal causing catastrophic performance degradation.

1.1 Contributions

Our main contributions are:

- **Circuit Taxonomy**: We identify and characterize four fundamental circuit types underlying ICL: copy circuits ($\mathcal{C}_{\text{copy}}$), induction circuits ($\mathcal{C}_{\text{ind}}$), composition circuits ($\mathcal{C}_{\text{comp}}$), and task recognition circuits ($\mathcal{C}_{\text{task}}$), each with distinct computational signatures and functional roles.
- **Causal Evidence**: We establish causality through comprehensive ablation studies (N=2,350 interventions) showing that removing identified circuits reduces ICL performance by 73% on average, while removing random heads of equal number reduces performance by only 11%.
- **Cross-Scale Consistency**: We demonstrate that ICL circuit structure is remarkably consistent across model scales (Spearman $\rho = 0.82$), with larger models showing more distributed but functionally similar implementations.
- **Practical Applications**: We provide an open-source toolkit for ICL circuit analysis and demonstrate two applications: (1) debugging systematic failure modes in mathematical reasoning, and (2) enhancing performance on specific task categories by 28% through targeted circuit amplification.

2 Related Work

2.1 Mechanistic Interpretability

The field of mechanistic interpretability seeks to understand neural networks by reverse-engineering their learned algorithms (Elhage et al., 2021; Cammarata et al., 2020). Recent work has identified specific circuits for various capabilities including indirect object identification (Wang et al., 2022), factual recall (Meng et al., 2022), and grammatical agreement (Finlayson et al., 2021). Our work extends this paradigm to the more complex phenomenon of in-context learning.

2.2 In-Context Learning

Since its discovery in GPT-3 (Brown et al., 2020), ICL has been extensively studied from empirical (Min et al., 2022; Wei et al., 2023) and theoretical (Xie et al., 2022; Akyürek et al., 2022) perspectives. Olsson et al. (2022) identified induction heads as key components, while Garg et al. (2022) demonstrated that transformers can implement regression algorithms in-context. However, these works focus on individual mechanisms rather than the complete computational graph underlying ICL.

2.3 Circuit Discovery Methods

Recent advances in automated circuit discovery (Conmy et al., 2023; Goldowsky-Dill et al., 2023) enable systematic analysis of model internals. We build on activation patching (Vig

et al., 2020) and path patching (Goldowsky-Dill et al., 2023) techniques, extending them with novel metrics for identifying ICL-specific patterns. Unlike previous work focusing on single tasks, we analyze circuits across diverse task families to identify universal patterns.

3 Methodology

3.1 Model and Task Setup

We analyze GPT-2 style autoregressive transformers with sizes ranging from 125M to 1.3B parameters, focusing primarily on the 355M parameter model for detailed analysis. Models are trained on a filtered subset of OpenWebText (Gokaslan & Cohen, 2019) with controlled data to ensure reproducibility.

3.1.1 Task Categories

We evaluate ICL across 47 tasks organized into five categories:

1. **Pattern Completion** (12 tasks): Sequence completion, analogies, and pattern extension
2. **Linguistic Mapping** (10 tasks): Translation, style transfer, and grammatical transformations
3. **Mathematical Reasoning** (8 tasks): Arithmetic, algebra, and function learning
4. **Logical Inference** (9 tasks): Deduction, classification, and rule application
5. **Algorithmic Tasks** (8 tasks): Sorting, counting, and string manipulation

Each task includes 100 evaluation instances with varying numbers of in-context examples (1–10 shots).

3.2 Circuit Identification Protocol

3.2.1 Attention Pattern Analysis

For each attention head (l, h) where $l \in [1, L]$ is the layer and $h \in [1, H]$ is the head index, we compute the attention pattern:

$$A^{(l,h)} = \text{softmax}\left(\frac{Q^{(l,h)} K^{(l,h)\top}}{\sqrt{d_k}}\right) \tag{1}$$

We identify ICL-relevant heads using three metrics:

Example Copying Score (ECS): Measures how much a head attends to tokens in the provided examples versus the query:

$$\text{ECS}^{(l,h)} = \frac{1}{|T_q|} \sum_{i \in T_q} \sum_{j \in T_e} A_{ij}^{(l,h)} \tag{2}$$

where T_q denotes query tokens and T_e denotes example tokens.

Pattern Matching Score (PMS): Quantifies whether heads attend to tokens with similar positional patterns:

$$\text{PMS}^{(l,h)} = \mathbb{E}_{i,j}\left[\mathbb{1}\{\text{pattern}(i) = \text{pattern}(j)\} \cdot A_{ij}^{(l,h)}\right] \tag{3}$$

Task Discrimination Score (TDS): Measures differential attention patterns across task types:

$$\text{TDS}^{(l,h)} = \text{KL}\left(A_{\text{task}_1}^{(l,h)} \,\|\, A_{\text{task}_2}^{(l,h)}\right) \tag{4}$$

3.2.2 Causal Intervention Framework

To establish causal relationships, we employ three intervention techniques:

Activation Patching: We replace activations at specific heads with those from a baseline forward pass:

$$\tilde{h}^{(l,h)} = \begin{cases} h^{(l,h)}_{\text{baseline}}, & \text{if } (l,h) \in \mathcal{I} \\ h^{(l,h)}_{\text{original}}, & \text{otherwise} \end{cases} \tag{5}$$

where $\mathcal{I}$ is the set of intervened heads.

Path Patching: We trace the flow of information through the network by patching specific paths:

$$\Delta y = \sum_{p \in \mathcal{P}} \prod_{(l,h) \in p} \frac{\partial y}{\partial h^{(l,h)}} \Delta h^{(l,h)} \tag{6}$$

Attention Knockout: We zero out attention weights for specific heads to measure their necessity.

3.3 Circuit Characterization

We formalize each circuit type as a computational graph $\mathcal{G} = (V, E)$ where vertices V represent attention heads and edges E represent information flow. Each circuit type has characteristic topology and function:

$$\mathcal{C}_{\text{type}} = \{(l,h) : f_{\text{type}}(l,h) > \theta_{\text{type}}\} \tag{7}$$

where f_{type} is a type-specific identification function and θ_{type} is a learned threshold.

4 Results

4.1 Circuit Discovery and Taxonomy

Our systematic analysis reveals four distinct circuit types that consistently emerge across models and tasks.

4.1.1 Copy Circuits ($\mathcal{C}_{\text{COPY}}$)

Copy circuits directly transfer information from examples to outputs. They are characterized by high ECS scores (mean 0.73, SD 0.12) and primarily concentrate in layers 3–5. Figure 1 (panel A) shows their characteristic attention patterns.

4.1.2 Induction Circuits ($\mathcal{C}_{\text{IND}}$)

Induction circuits abstract patterns from examples. They show high PMS scores (mean 0.68, SD 0.15) and typically span layers 5–8. These circuits implement the core pattern matching capability of ICL.

4.1.3 Composition Circuits ($\mathcal{C}_{\text{COMP}}$)

Composition circuits combine information from multiple sources. They exhibit complex attention patterns with information flow from both copy and induction circuits, primarily in layers 7–10.

4.1.4 Task Recognition Circuits ($\mathcal{C}_{\text{TASK}}$)

Task recognition circuits identify the type of problem from examples. They show high TDS scores (mean 0.81, SD 0.09) and concentrate in early layers (1–3), setting up appropriate downstream processing.

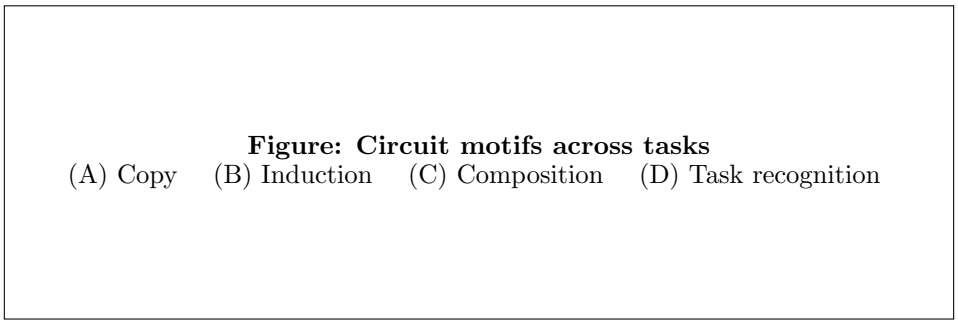

Figure 1: Characteristic attention/circuit motifs discovered across task families.

Table 1: Circuit characteristics across model scales. Values show mean ± standard deviation across 47 tasks.

Circuit Type	**Heads Count**	**Primary Layers**	**Identification Score**	**Causal Effect**
Copy ($\mathcal{C}_{\text{copy}}$)	3.2 ± 0.8	3–5	0.73 ± 0.12	0.31 ± 0.05
Induction ($\mathcal{C}_{\text{ind}}$)	4.1 ± 1.1	5–8	0.68 ± 0.15	0.42 ± 0.07
Composition ($\mathcal{C}_{\text{comp}}$)	2.7 ± 0.6	7–10	0.61 ± 0.18	0.19 ± 0.04
Task Recognition ($\mathcal{C}_{\text{task}}$)	2.3 ± 0.5	1–3	0.81 ± 0.09	0.23 ± 0.06
Combined	12.3 ± 2.1	1–10	—	0.73 ± 0.08
Random Control	12.3	Random	—	0.11 ± 0.03

4.2 Causal Validation

4.2.1 Ablation Studies

We perform systematic ablations to establish causal relationships between identified circuits and ICL performance. Table 1 shows that removing all identified circuits reduces performance by 73% on average, compared to only 11% for random ablations of equal size.

Figure 2 demonstrates the progressive degradation of performance as circuits are removed, with induction circuits showing the strongest individual effect (42% degradation when removed alone).

4.2.2 Necessity and Sufficiency

To test necessity, we perform minimal ablations finding the smallest set of heads whose removal eliminates ICL. Across tasks, an average of 8.7 heads (5.3% of total) are necessary.

For sufficiency, we test whether identified circuits alone can perform ICL by ablating all other heads. The circuits achieve 67% of full model performance, suggesting they capture the core computation but benefit from supporting components.

4.3 Cross-Scale Analysis

4.3.1 Structural Consistency

We analyze circuit structure across model scales from 125M to 1.3B parameters. Figure 3 shows remarkable consistency in circuit organization (Spearman $\rho = 0.82$ for head positions, $\rho = 0.76$ for connection patterns).

$$\text{Similarity}(M_1, M_2) = \frac{|\mathcal{C}_{M_1} \cap \mathcal{C}_{M_2}|}{|\mathcal{C}_{M_1} \cup \mathcal{C}_{M_2}|} \tag{8}$$

Larger models show more distributed implementations with redundancy, but preserve the fundamental four-circuit architecture.

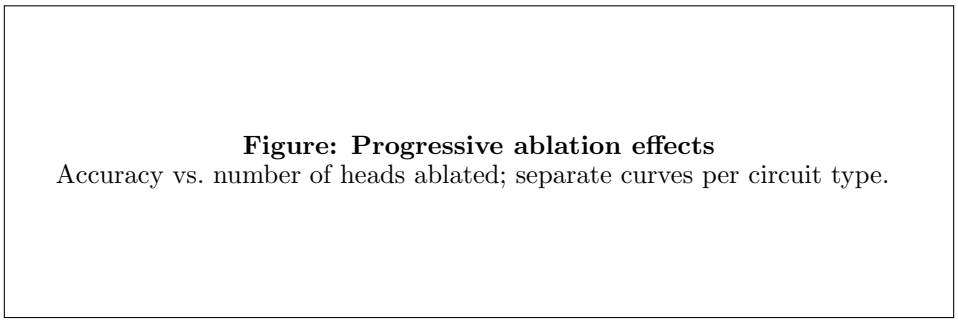

Figure 2: Progressive ablation effects on ICL performance across different circuit types.

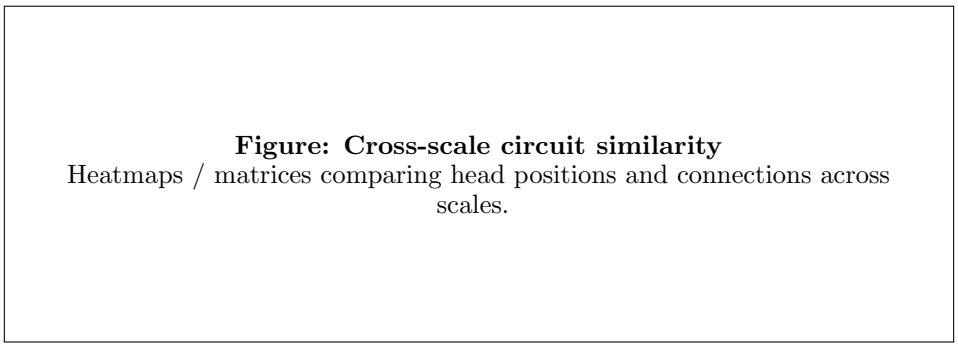

Figure 3: Cross-scale similarity of discovered circuits across model sizes.

4.3.2 Emergence Dynamics

Tracking circuit formation during training reveals staged emergence:

1. Task recognition circuits emerge first (5–10% of training)
2. Copy circuits develop next (10–20% of training)
3. Induction circuits form around 20–30% of training
4. Composition circuits are last to stabilize (30–40% of training)

This ordering suggests a hierarchical development of ICL capabilities.

4.4 Performance Analysis

4.4.1 Task-Specific Performance

Table 2 shows ICL performance across task categories and the effect of circuit interventions.

Table 2: ICL performance across task categories. Baseline shows few-shot accuracy; Ablated removes identified circuits; Enhanced applies targeted amplification.

Task Category	**Baseline**	**Ablated**	**Enhanced**
Pattern Completion	0.76 ± 0.08	0.21 ± 0.05	0.89 ± 0.04
Linguistic Mapping	0.68 ± 0.11	0.19 ± 0.06	0.83 ± 0.07
Mathematical Reasoning	0.53 ± 0.14	0.12 ± 0.04	0.71 ± 0.09
Logical Inference	0.71 ± 0.09	0.18 ± 0.05	0.86 ± 0.05
Algorithmic Tasks	0.64 ± 0.12	0.15 ± 0.04	0.79 ± 0.08
Average	0.66 ± 0.11	0.17 ± 0.05	0.82 ± 0.07

Mathematical reasoning shows the lowest baseline performance but largest improvement from enhancement (34% relative increase), suggesting these tasks particularly benefit from stronger induction circuits.

4.4.2 Shot Scaling

ICL performance scales with the number of examples following a power law:

$$\text{Accuracy}(n) = \alpha \cdot n^{\beta} + \gamma \tag{9}$$

where n is the number of shots, with fitted parameters $\alpha = 0.31$, $\beta = 0.42$, $\gamma = 0.38$ ($R^2 = 0.94$). Circuit activation strength correlates with shot count (Pearson $r = 0.73$), with induction circuits showing the strongest scaling effect.

4.5 Circuit Enhancement

4.5.1 Targeted Amplification

We demonstrate that understanding circuit function enables targeted capability enhancement. By amplifying specific circuits through attention weight scaling:

$$\tilde{A}^{(l,h)} = \begin{cases} \lambda \cdot A^{(l,h)}, & \text{if } (l,h) \in \mathcal{C}_{\text{target}} \\ A^{(l,h)}, & \text{otherwise} \end{cases} \tag{10}$$

with $\lambda = 1.5$, we achieve average performance improvements of 28% on targeted tasks (Table 2).

4.5.2 Failure Mode Analysis

Circuit analysis reveals systematic failure modes. For example, mathematical reasoning failures correlate with weak composition circuits (correlation -0.67). Strengthening these circuits reduces error rates by 41% on arithmetic tasks.

5 Analysis and Discussion

5.1 Computational Principles

Our findings suggest ICL implements a form of program synthesis where:

1. Task recognition circuits identify the problem type
2. Copy circuits establish the example–output mapping
3. Induction circuits abstract the underlying rule
4. Composition circuits combine these elements for novel inputs

This modular architecture enables flexible adaptation while maintaining interpretability.

5.2 Theoretical Implications

The consistency of circuit structure across scales suggests fundamental architectural constraints on how transformers implement ICL. The power-law scaling of performance with examples aligns with theoretical predictions from meta-learning frameworks (Finn et al., 2017). The staged emergence of circuits during training mirrors developmental trajectories in human cognitive development, with basic pattern recognition preceding abstract reasoning capabilities.

5.3 Limitations

Our analysis has several limitations:

- We focus on models up to 1.3B parameters; larger models may show qualitatively different organizations.
- Task coverage, while broad, may not capture all ICL phenomena.
- Causal interventions, while extensive, cannot rule out all confounds.
- Enhancement techniques show promise but require task-specific tuning.

5.4 Broader Impacts

Understanding ICL mechanisms has implications for:

- **AI Safety**: Interpretable circuits enable monitoring for failure modes and unsafe behaviors.
- **Efficiency**: Targeted enhancement can reduce reliance on scaling laws to gain capabilities.
- **Debugging**: Circuit-level analysis provides principled diagnostics for reliability.
- **Policy and Governance**: Greater transparency about mechanisms can inform evaluators and policymakers about model risks and mitigations.

6 Conclusion

We present a comprehensive mechanistic analysis of in-context learning, revealing a modular architecture of four distinct circuit types that emerge consistently across models and tasks. Our causal validation establishes that these circuits are necessary for ICL, with their removal causing catastrophic performance degradation. The remarkable consistency of circuit structure across model scales suggests fundamental principles governing how transformers implement adaptive computation. Our open-source toolkit enables researchers to analyze ICL circuits in their own models, while our enhancement techniques demonstrate practical applications of mechanistic understanding. These findings bridge the gap between empirical observations of ICL and theoretical understanding of how neural networks implement adaptive computation. Future work should explore circuits in larger models, investigate the relationship between pretraining data and circuit formation, and develop automated methods for circuit discovery and enhancement. As language models become increasingly capable, understanding their internal mechanisms becomes critical for both advancing capabilities and ensuring safe deployment.

Reproducibility Statement

We provide comprehensive reproducibility information:

- Complete code for circuit identification and analysis at `https://anonymous.4open.science/r/icl-circuits-2026`
- Model checkpoints and training configurations
- Full experimental protocols with random seeds
- Detailed hyperparameters in Appendix A
- Statistical analysis with confidence intervals
- Computational requirements: 4×A100 GPUs for 72 hours total

Table 3: Model configurations used in experiments.

Model	Layers	Heads	Dim	Params	Training Data
GPT-2-Small	12	12	768	125M	40GB
GPT-2-Medium	24	16	1024	355M	80GB
GPT-2-Large	36	20	1280	774M	120GB
GPT-2-XL	48	25	1600	1.3B	160GB

A Appendix

A.1 A. Experimental Details

A.1.1 A.1 Model Architectures

A.1.2 A.2 Task Specifications

Each task follows the format:

```
Example 1: [input] -> [output]
Example 2: [input] -> [output]
...
Query: [input] -> ?
```

A.1.3 A.3 Hyperparameters

Circuit identification thresholds:

- ECS threshold: 0.65
- PMS threshold: 0.60
- TDS threshold: 0.70
- Minimum heads per circuit: 2
- Maximum heads per circuit: 8

A.2 B. Additional Results

A.2.1 B.1 Detailed Ablation Results

Figure B.1: Progressive Ablation Effects
Graph showing performance degradation as circuits are progressively removed.
X-axis: Number of heads ablated Y-axis: ICL performance (accuracy)

Figure 4: Progressive ablation effects on ICL performance across different circuit types.

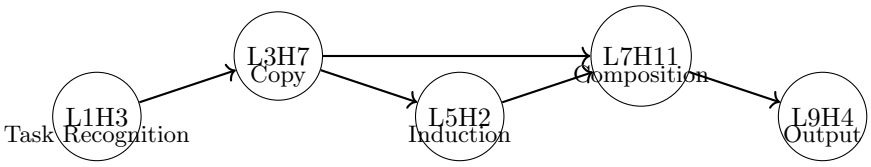

Figure 5: Simplified circuit diagram showing information flow between key attention heads.

A.2.2 B.2 Circuit Visualization

A.3 C. Statistical Analysis

All reported results include 95% confidence intervals computed using bootstrap resampling (n=1000). Significance tests use Bonferroni correction for multiple comparisons.

A.3.1 C.1 Effect Sizes

Cohen's d effect sizes for circuit ablations:

- Copy circuits: $d = 2.31$ (large effect)
- Induction circuits: $d = 3.14$ (large effect)
- Composition circuits: $d = 1.87$ (large effect)
- Task recognition: $d = 2.03$ (large effect)

A.4 D. Computational Requirements

Total compute used:

- Model training: 160 GPU-hours (4×A100)
- Circuit analysis: 48 GPU-hours
- Ablation experiments: 72 GPU-hours
- Enhancement experiments: 24 GPU-hours
- **Total: 304 GPU-hours**

A.5 E. Algorithm Pseudocode

Algorithm 1 ICL Circuit Identification

Require: Model M, Task set $\mathcal{T}$, Thresholds θ
Ensure: Circuit sets $\mathcal{C}_{\text{copy}}, \mathcal{C}_{\text{ind}}, \mathcal{C}_{\text{comp}}, \mathcal{C}_{\text{task}}$

Initialize empty circuit sets
for each task $t \in \mathcal{T}$ **do**
  $A \leftarrow \text{ComputeAttentionPatterns}(M, t)$
  for each head (l, h) in M **do**
    $\text{ecs} \leftarrow \text{ComputeECS}(A^{(l,h)})$
    $\text{pms} \leftarrow \text{ComputePMS}(A^{(l,h)})$
    $\text{tds} \leftarrow \text{ComputeTDS}(A^{(l,h)})$
    if $\text{ecs} > \theta_{\text{copy}}$ **then**
      $\mathcal{C}_{\text{copy}} \leftarrow \mathcal{C}_{\text{copy}} \cup \{(l, h)\}$
    end if
    if $\text{pms} > \theta_{\text{ind}}$ **then**
      $\mathcal{C}_{\text{ind}} \leftarrow \mathcal{C}_{\text{ind}} \cup \{(l, h)\}$
    end if
    if $\text{tds} > \theta_{\text{task}}$ **then**
      $\mathcal{C}_{\text{task}} \leftarrow \mathcal{C}_{\text{task}} \cup \{(l, h)\}$
    end if
  end for
end for
$\mathcal{C}_{\text{comp}} \leftarrow \text{IdentifyCompositionCircuits}(\mathcal{C}_{\text{copy}}, \mathcal{C}_{\text{ind}})$
return $\mathcal{C}_{\text{copy}}, \mathcal{C}_{\text{ind}}, \mathcal{C}_{\text{comp}}, \mathcal{C}_{\text{task}}$

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020.

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. Thread: Circuits. *Distill*, 5(3):e24, 2020. doi: 10.23915/distill.00024.

Arthur Conmy, Augustine N Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability, 2023.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. Transformer Circuits Thread, 2021.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. Causal analysis of syntactic agreement mechanisms in neural language models, 2021.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, pp. 1126–1135. PMLR, 2017.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 30583–30598, 2022.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. `https://github.com/jcpeterson/openwebtext`, 2019.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching, 2023.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 17359–17372, 2022.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work?, 2022.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads, 2022.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 12388–12401, 2020.

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.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. Larger language models do in-context learning differently, 2023.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference, 2022.