CIRCUIT DISTILLATION FOR MATH REASONING: ALIGNING COMPUTATIONAL MECHANISMS IN LARGE LANGUAGE MODELS

ESHAN SINGHAL [ESINGHAL@SEAS], AUDHAV DURAI [AUDHAV@WHARTON],
VEDANT GAUR [VEDANTG@WHARTON], PRANEEL VARSHNEY [PVARSH@SEAS]

ABSTRACT. *Circuit Distillation* is a mechanism-level method for transferring knowledge from a large teacher model to a smaller student model by aligning internal computational circuits rather than matching outputs, as is done in traditional knowledge distillation. We apply this approach to arithmetic reasoning in the Llama 3 family with a novel method that has a two-stage pipeline. First, we discover arithmetic-relevant circuits in the teacher using a classifier-based framework that clusters arithmetic problems into latent classes and learns sparse, mutually distinct, and functionally selective neuron-level masks for each class. Second, we distill these circuits into the student by enforcing mechanism-level alignment through representational similarity objectives.

[Add 2-3 sentences of results here]

1. INTRODUCTION

Large language models (LLMs) exhibit strong arithmetic and symbolic reasoning abilities when scaled to tens of billions of parameters [1]. However, deploying such models is often infeasible due to computational and memory constraints. Knowledge distillation aims to address this by transferring capabilities from large teacher models to smaller student models, but conventional approaches focus on matching outputs rather than internal computation. As a result, distilled models often produce correct answers while relying on brittle heuristics rather than the teacher's underlying algorithmic mechanisms [3].

Mechanistic interpretability suggests that many reasoning behaviors in transformers are implemented by sparse, functionally specialized subcircuits composed of particular neurons and attention heads [4]. This motivates the question: *can we identify such circuits and distill them directly into smaller models?* If so, student models could inherit not just surface-level behavior but the teacher's internal reasoning process.

In this work, we focus on arithmetic reasoning and propose a neuron-level circuit discovery and distillation framework. We introduce a method that (i) clusters arithmetic problems into latent classes, (ii) learns sparse neuron masks for each class using a structured optimization objective, and (iii) uses these masks as the basis for circuit-level distillation that leverages representational similarity alignment between teacher and student [8, 9]. Our approach emphasizes generality. That is, as long as the number of latent classes is sufficiently large to capture meaningful structure, the precise value of this hyperparameter is not critical.

1.1. Contributions.

- We propose a neuron-level circuit discovery framework for arithmetic reasoning that jointly clusters problems and learns sparse neuron masks corresponding to each cluster.
- We introduce a multi-term objective that enforces sparsity, balanced usage, and orthogonality between discovered circuits, yielding interpretable and functionally distinct neuron groups.
- We apply this framework to both 1B- and 8B-parameter transformer models, demonstrating consistent circuit structure across scales.
- We outline how the discovered circuits can be integrated into a circuit distillation pipeline for mechanism-level transfer using representational similarity objectives.

2. BACKGROUND

2.1. Transformers and Arithmetic. Transformers process input sequences through layers of self-attention and feedforward networks (FFNs), producing intermediate activations that can be interpreted as contributing to the model's computation [4]. Arithmetic problems typically consist of fixed-length token sequences (e.g., three tokens for operands and two for operators), yielding structured activation patterns across layers.

2.2. **Neuron Masks as Circuits.** We model a circuit as a *neuron mask*: a vector of weights applied multiplicatively to neuron outputs in FFN layers. This view aligns with research that circuits in transformers can be viewed as sparse subgraphs of the model's computational graph responsible for particular behaviors [2, 4]. Learning such masks facilitates both interpretability and targeted interventions.

2.3. **Representational Similarity.** Matching student and teacher representations using metrics invariant to orthogonal transformations, such as Centered Kernel Alignment (CKA), has proven effective for comparing internal states of models with different architectures [8, 9]. We leverage this in the distillation objective to align circuit activations across student and teacher.

3. Related Work

Knowledge Distillation. Traditional distillation matches teacher and student output distributions and, in some cases, intermediate representations. However, these approaches do not explicitly transfer internal reasoning mechanisms. Chain-of-thought distillation has been proposed to include teacher rationales, but students may imitate style without acquiring causal computation [3].

Mechanistic Interpretability and Circuits. Mechanistic interpretability aims to decompose networks into interpretable components such as attention heads and neurons implementing specific functions. The transformer circuits paradigm formalizes this view, identifying subnetworks responsible for tasks like induction and arithmetic reasoning [4]. Attribution Patching provides a scalable approximation for estimating component importance [5, 6].

Similarity Metrics. Centered Kernel Alignment (CKA) has emerged as a robust method for comparing representations across models, due to its invariance to invertible linear transformations and stability in high dimensions [8, 9].

4. Approach

Our circuit discovery framework consists of three components: problem encoding and clustering, neuron mask generation, and a multi-term optimization objective that enforces interpretability and functional separation.

4.1. **Problem Encoding and Latent Classes.** Given an arithmetic problem x_i, we compute a fixed-dimensional problem encoding using intermediate transformer activations. This encoding is passed to a K-class classifier, where K is a user-specified hyperparameter controlling the granularity of discovered circuits. The classifier's output distribution is sampled via Gumbel-Softmax with a straight-through estimator to facilitate end-to-end gradients while yielding discrete class assignments.

4.2. **Neuron Mask Generator.** Conditioned on the sampled class, a two-layer feedforward network produces a neuron mask $\mathbf{m}_k \in (0, 1)^D$, where D is the number of neurons in the target FFN layers. Each mask weight modulates the corresponding neuron activation via elementwise multiplication. Masks are shared across all problems assigned to the same class, encouraging each class to correspond to a consistent computational subcircuit.

4.3. **Objective Function.** We introduce the following notation:

- K: number of latent problem classes.
- $i \in \{1, \ldots, N_k\}$: index over problems assigned to class k.
- $d \in \{1, \ldots, D\}$: neuron index.
- $a_{i,k,d}$: activation of neuron d for problem i in class k.
- $m_{k,d}$: neuron mask weight for neuron d in class k.
- $\mathbf{M}_k \in \mathbb{R}^D$: mask vector for class k.
- f_k: empirical probability that class k is selected.
- π: target average mask activation (e.g., 0.1).

The total loss is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{sim}} + \lambda_2 \mathcal{L}_{\text{usage}} + \lambda_3 \mathcal{L}_{\text{sparsity}} + \lambda_4 \mathcal{L}_{\text{KL}} + \lambda_5 \mathcal{L}_{\text{mask-sim}}.$$

4.3.1. *Similarity Loss.*

$$\mathcal{L}_{\text{sim}} = -\frac{1}{K} \sum_{k=1}^{K} \frac{1}{N_k(N_k - 1)} \sum_{d=1}^{D} \left(\sum_{i=1}^{N_k} \|a_{i,k,d}\| \right)^2 .$$

4.3.2. *Usage Entropy.*

$$\mathcal{L}_{\text{usage}} = -\sum_{k=1}^{K} f_k \log f_k.$$

4.3.3. *Sparsity Loss.*

$$\mathcal{L}_{\text{sparsity}} = -\frac{1}{K}\sum_{k=1}^{K}\frac{1}{D}\sum_{d=1}^{D}\left[m_{k,d}\log m_{k,d} + (1-m_{k,d})\log(1-m_{k,d})\right].$$

4.3.4. *KL Regularization.*

$$\mathcal{L}_{\text{KL}} = \bar{m}\log\frac{\bar{m}}{\pi} + (1-\bar{m})\log\frac{1-\bar{m}}{1-\pi}, \quad \bar{m} = \frac{1}{KD}\sum_{k,d} m_{k,d}.$$

4.3.5. *Mask Orthogonality Loss.*

$$\mathcal{L}_{\text{mask-sim}} = \frac{1}{K(K-1)}\sum_{k_1=1}^{K}\sum_{\substack{k_2=1\\ k_2\neq k_1}}^{K} \mathbf{M}_{k_1}^{\top}\mathbf{M}_{k_2}.$$

4.4. **Circuit Discovery Model Architecture.** We learn a latent-class circuit decomposition over arithmetic problems and use it to generate sparse neuron-level masks over MLP activations. Given an equation x (e.g., $12 + 34 = 46$), we parse the operands and result (o_1, o_2, r) and embed them with a lightweight *ProblemEncoder* that concatenates learned embeddings for o_1, o_2, and r into a fixed-dimensional vector $\phi(x)$.

A small MLP classifier maps $\phi(x)$ to logits over K latent classes, and we sample a discrete class assignment using straight-through Gumbel-Softmax. Conditioned on the sampled class, we generate neuron masks separately for the 1B and 8B models. Each mask generator is a compact network (class embedding → linear → sigmoid) that outputs a mask vector $\mathbf{m} \in (0,1)^D$, where $D = \texttt{intermediate\_size} \times \texttt{num\_hidden\_layers}$ corresponds to flattened MLP neurons across all transformer blocks. The mask is applied multiplicatively to the stacked per-layer MLP activations.

We optimize a multi-term objective that (i) increases within-class similarity of masked activations, (ii) encourages sparse/binary masks via entropy and a global Bernoulli KL-to-prior term, (iii) discourages overlap between class masks via mean pairwise cosine similarity, and (iv) encourages balanced class usage via an entropy bonus.

4.4.1. *Circuit Discovery Training Dynamics.* We analyze the optimization behavior of the circuit discovery objective to verify that it produces sparse, distinct, and functionally meaningful neuron masks. First, the sparsity-related terms jointly encourage masks with a controlled activation budget: the binary-entropy sparsity loss drives mask entries toward near-binary values, while the KL regularization term constrains the global mean activation rate toward a Bernoulli prior with $\pi = 0.1$. As training progresses, these terms decrease in tandem, indicating convergence to sparse masks with approximately 10% active neurons.

Second, we examine the interaction between representativeness and separation of discovered circuits. The within-class similarity objective encourages masked activations corresponding to the same latent class to be consistent, while the mask cosine similarity penalty discourages overlap between class-conditioned masks. Over training, similarity improves as mask cosine similarity decreases, suggesting that the learned circuits become both representative of their assigned problem classes and increasingly orthogonal to one another. See Appendix B for training curves.

References

[1] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and others. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.

[2] Somin Wadhwa, Silvio Amir, and Byron C. Wallace. Circuit Distillation. *arXiv preprint arXiv:2509.25002*, 2025.

[3] Somin Wadhwa, Silvio Amir, and Byron C. Wallace. Investigating mysteries of CoT-augmented distillation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6071–6086, 2024.

[4] Nelson Elhage, Neel Nanda, Catherine Olsson, and others. A Mathematical Framework for Transformer Circuits. *Transformer Circuits Thread*, 2021.

[5] Neel Nanda. Attribution Patching: Activation Patching at Industrial Scale. 2023.

[6] János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. AtP*: An efficient and scalable method for localizing LLM behaviour to components. *arXiv preprint arXiv:2403.00745*, 2024.

[7] Aaron Mueller and others. MIB: A Mechanistic Interpretability Benchmark. *arXiv preprint arXiv:2504.13151*, 2025.

[8] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. In *International Conference on Machine Learning*, 2019.

[9] Alex Williams and others. An Equivalence Between Representational Similarity Analysis and Centered Kernel Alignment. *Cognitive Computational Neuroscience*, 2025.

APPENDIX A. ADDITIONAL FIGURES

A.1. **Circuit Discovery Architecture.**

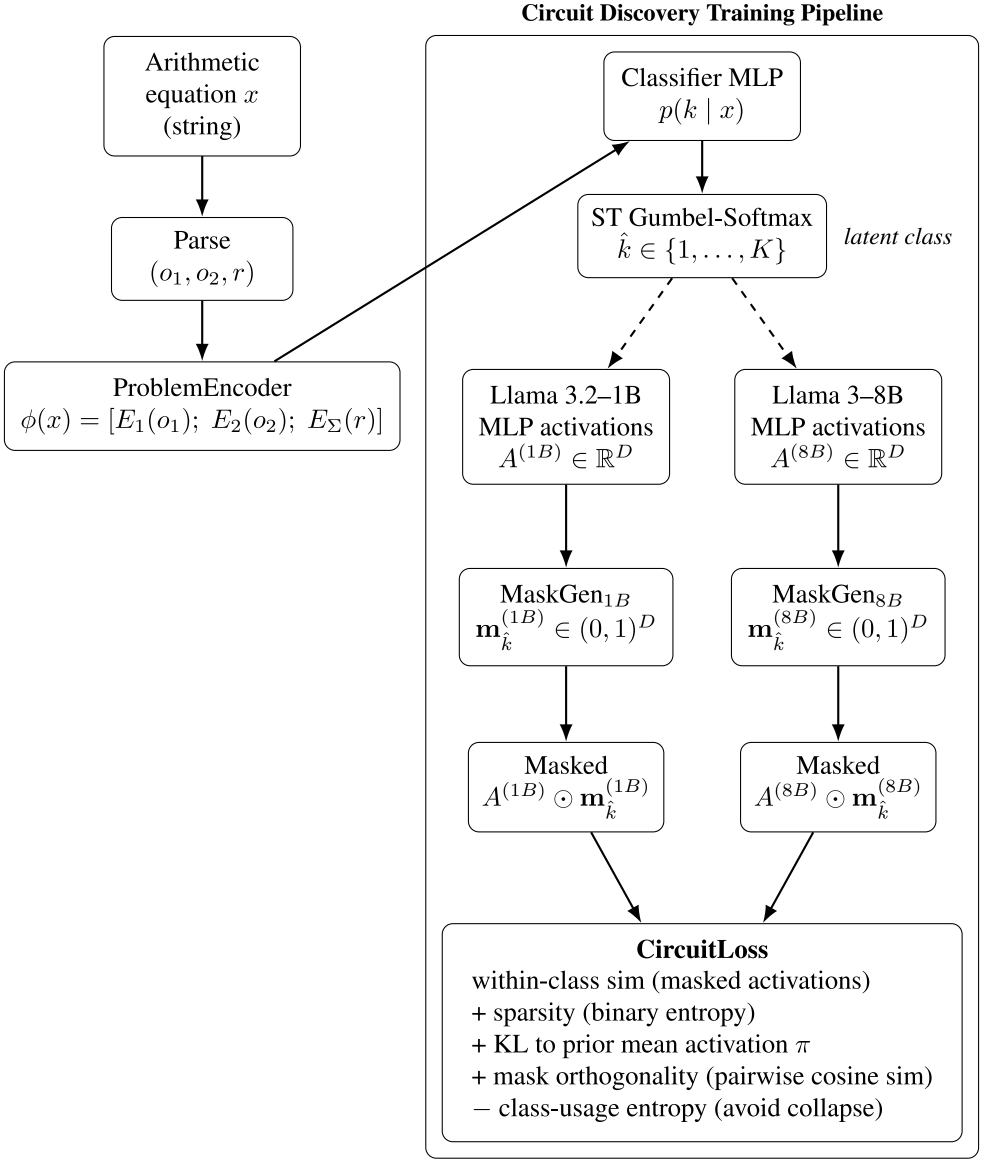
Circuit Discovery Training Pipeline
Arithmetic equation x (string)
Parse (o_1, o_2, r)
ProblemEncoder $\phi(x) = [E_1(o_1);\ E_2(o_2);\ E_\Sigma(r)]$
Classifier MLP $p(k \mid x)$
ST Gumbel-Softmax $\hat{k} \in \{1, \ldots, K\}$
latent class
Llama 3.2–1B MLP activations $A^{(1B)} \in \mathbb{R}^D$
Llama 3–8B MLP activations $A^{(8B)} \in \mathbb{R}^D$
MaskGen$_{1B}$ $\mathbf{m}_{\hat{k}}^{(1B)} \in (0,1)^D$
MaskGen$_{8B}$ $\mathbf{m}_{\hat{k}}^{(8B)} \in (0,1)^D$
Masked $A^{(1B)} \odot \mathbf{m}_{\hat{k}}^{(1B)}$
Masked $A^{(8B)} \odot \mathbf{m}_{\hat{k}}^{(8B)}$
CircuitLoss
within-class sim (masked activations)
+ sparsity (binary entropy)
+ KL to prior mean activation π
+ mask orthogonality (pairwise cosine sim)
− class-usage entropy (avoid collapse)

FIGURE 1. Circuit discovery architecture. Problems are embedded and assigned to a latent class via a straight-through Gumbel-Softmax classifier. The sampled class selects a class-conditioned neuron mask for each teacher model (1B and 8B), which gates flattened MLP activations. Training optimizes a multi-term objective encouraging within-class functional similarity, sparse masks, distinct circuits, and balanced class usage.

Appendix B. Circuit Discovery Training Dynamics

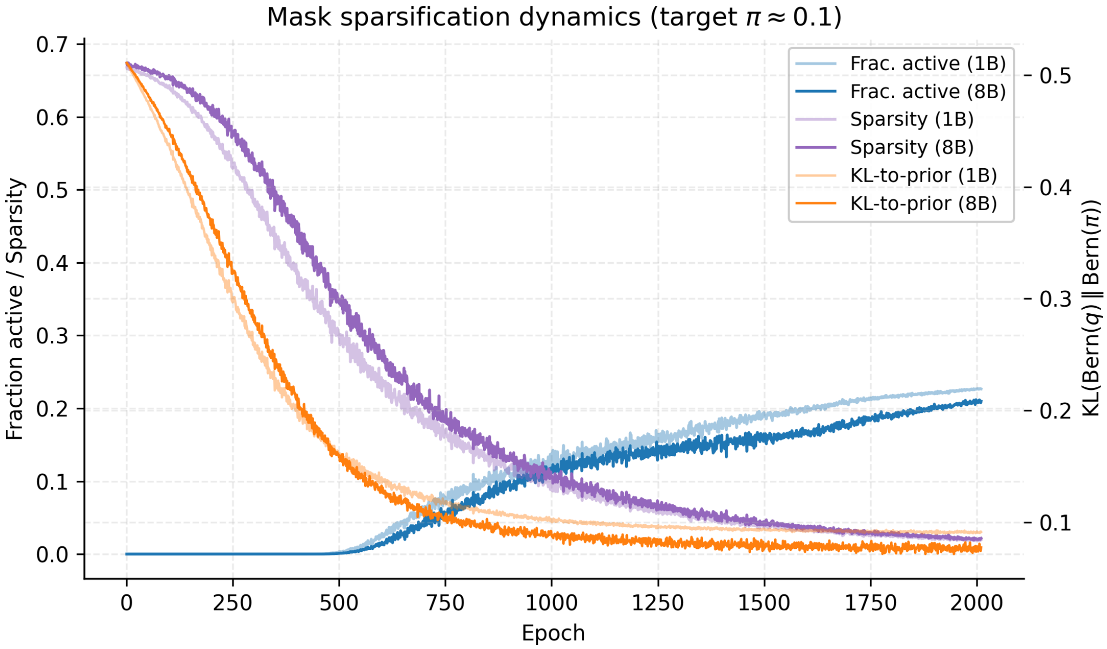

FIGURE 2. Circuit discovery training curves related to sparsity regularization (binary-entropy sparsity loss and KL-to-prior mean activation). These curves support the discussion in Section 4.5.

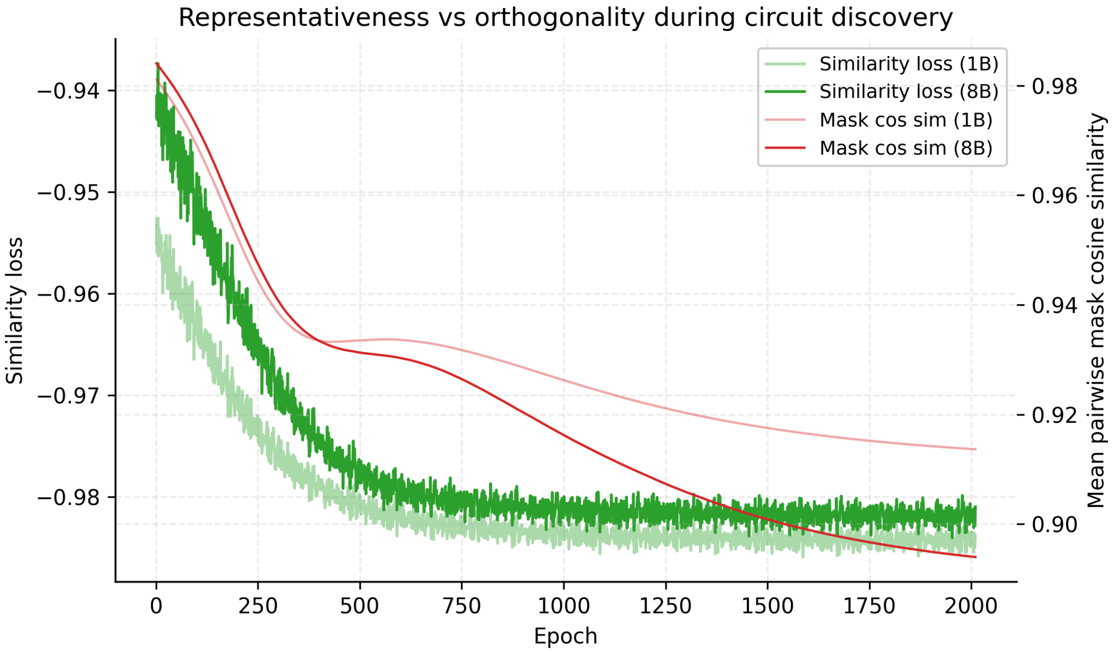

FIGURE 3. Circuit discovery training curves related to representativeness (within-class similarity) and separation (mask orthogonality / cosine similarity). These curves support the discussion in Section 4.5.