Circuit Distillation for Math Reasoning: Aligning Computational Mechanisms in Large Language Models

Research Report

December 15, 2025

Abstract

The prevailing paradigm in the compression of Large Language Models (LLMs)—knowledge distillation—has historically relied on behavioral mimicry, wherein a student model is trained to replicate the final output distributions of a larger teacher model. While effective for preserving surface-level performance on in-distribution tasks, this approach treats the internal reasoning processes of the teacher as a black box, often failing to transfer the robust generalization capabilities and safety mechanisms embedded in the teacher's latent computations. This report articulates a comprehensive framework for **Circuit Distillation**, a novel methodology designed to align the internal computational subgraphs ("circuits") of a student model with those of a teacher, specifically targeting mathematical reasoning capabilities within the Llama 3 architecture. By leveraging recent advances in mechanistic interpretability—specifically Attribution Patching and the Mechanistic Interpretability Benchmark (MIB)—we propose a theoretical and experimental pipeline to identify, map, and distill the specific attention heads and MLPs responsible for arithmetic operations. We review the efficacy of representational similarity metrics, such as Centered Kernel Alignment (CKA), in quantifying mechanism transfer and discuss the implications of moving from outcome-based to process-based distillation. This document serves as a foundational text for the validation of mechanism transfer in arithmetic reasoning, providing a blueprint for the construction of smaller, more robustly aligned models that reason for the right reasons.

1 Introduction

1.1 The Scaling Hypothesis and the Opacity Barrier

The trajectory of recent artificial intelligence research has been defined by the Scaling Hypothesis, which posits that performance on downstream tasks scales as a power law with respect to parameter count, dataset size, and compute (**?**). Models such as Llama 3 70B have demonstrated emergent capabilities in complex reasoning, coding, and mathematics that were not present in their smaller predecessors. However, the deployment of these massive models in real-world applications is severely constrained by computational latency and energy costs.

To address this, the field has turned to Knowledge Distillation (KD), a technique traditionally grounded in the minimization of the Kullback-Leibler (KL) divergence between the output logits of a teacher model and a student model. While KD has successfully produced smaller models that approximate the teacher's accuracy, it fundamentally suffers from an "opacity barrier." Standard KD incentivizes the student to match the *result* of the teacher's computation, but not the *process*. Consequently, student models frequently learn to produce correct answers via heuristic shortcuts or memorization—a phenomenon known as "Clever Hans" behavior—rather than implementing the robust, generalizable algorithms developed by the teacher (**?**).

1.2 The Mechanistic Turn in Interpretability

Parallel to the development of distillation techniques, the field of Mechanistic Interpretability has matured into a rigorous science aimed at reverse-engineering the algorithms implemented by neural network weights. We now understand that Transformers operate through "circuits"—sparse subgraphs of the model consisting of specific attention heads and MLP neurons that implement distinct functions (**?**). For instance, recent work has identified specific "induction heads" responsible for in-context learning and "arithmetic heads" that perform numerical operations through trigonometric manipulations of representations in the residual stream (**?**).

This granular understanding of model internals offers a new path for model compression. If we can identify the specific circuit C_T within a teacher model responsible for mathematical reasoning, we can theoretically force a student model to implement an analogous circuit C_S. This approach, termed **Circuit Distillation**, posits that aligning the internal computational mechanisms of the student with those of the teacher will result in superior out-of-distribution generalization and robustness compared to purely behavioral objectives (**?**).

1.3 Research Objectives and Scope

This report outlines the theoretical foundation, methodological framework, and experimental design for applying Circuit Distillation to the domain of mathematical reasoning. We focus specifically on the Llama 3 family of models, utilizing the 70B model as the teacher and the 8B model as the student (**?**).

Our primary objectives are:

1. **Circuit Identification:** To utilize scalable interpretability methods, specifically Attribution Patching (AtP), to localize the arithmetic circuits within the Llama 3 70B teacher model. We leverage the Mechanistic Interpretability Benchmark (MIB) to validate the fidelity of these circuits (**?**).

2. **Mechanism Alignment:** To formulate a loss function based on representational similarity—specifically Centered Kernel Alignment (CKA)—that enforces geometric alignment between the teacher's reasoning subspace and the student's, accommodating architectural differences such as Grouped Query Attention (GQA) (**?**).

3. **Rigorous Evaluation:** To move beyond accuracy metrics and evaluate the "faithfulness" and "completeness" of the distilled circuits using the MIB framework, ensuring that the student has truly internalized the teacher's algorithm (**?**).

2 Related Work

2.1 Knowledge Distillation: From Logits to Features

The evolution of Knowledge Distillation (KD) reflects a consistent drive to transfer richer information from teacher to student. The seminal work in KD focused on "response-based" knowledge, using the soft targets of the teacher's output layer to guide the student. While effective for classification, this approach is limited in generative tasks where the output space is vast and the reasoning chain is latent.

Subsequent "feature-based" distillation methods attempted to align intermediate activations, typically by minimizing the Mean Squared Error (MSE) between matched layers. However, these methods often fail to account for the fact that large and small models may encode the same information in orthogonal subspaces or at different depths. Furthermore, feature-based distillation typically aligns the *entire* residual stream, introducing noise from task-irrelevant features.

Most recently, Chain-of-Thought (CoT) augmented distillation has been proposed to transfer reasoning by including the teacher's step-by-step rationale in the training data. **?** investigated this approach and found a troubling result: while students trained on CoT data generate text that *looks* like reasoning, they often fail to utilize the causal mechanisms implied by that text. The student learns to mimic the style of the explanation without acquiring the underlying computational capability.

2.2 Mechanistic Interpretability: The Search for Circuits

The hypothesis that neural networks are composed of interpretable components has been validated through the discovery of circuits for specific tasks, such as Indirect Object Identification (IOI) and modular arithmetic. The "Mathematical Framework for Transformer Circuits" establishes the residual stream as a communication channel where attention heads read and write information (**?**).

Attribution Patching: A critical challenge in circuit discovery is scalability. "Activation Patching" (or causal tracing) is the gold standard for identifying important components, but it requires a separate forward pass for every component ablated, which is computationally prohibitive for models like Llama 3 70B. To solve this, **?** introduced **Attribution Patching (AtP)**, a gradient-based approximation that estimates the causal effect of every edge in the model in a single backward pass (**?**). AtP uses a first-order Taylor expansion to approximate the effect of patching, making it feasible to map circuits in industrial-scale models.

The MIB Benchmark: The lack of standardized evaluation has historically fragmented the interpretability field. The **Mechanistic Interpretability Benchmark (MIB)**, released in 2025, provides a rigorous testbed for evaluating circuit discovery methods (**?**). MIB includes verified ground-truth circuits for tasks like arithmetic addition and subtraction, allowing researchers to quantify the precision and recall of discovery techniques like AtP versus baselines.

2.3 Representational Similarity and CKA

To align the internal states of models with different architectures, we require a metric invariant to invertible linear transformations. **Centered Kernel Alignment (CKA)** has emerged as the robust standard for comparing neural representations (**?**). Unlike canonical correlation analysis (CCA), CKA is stable on high-dimensional data and can reliably identify correspondences between networks trained from different initializations. Recent theoretical work has shown that CKA is mathematically equivalent to specific forms of Representational Similarity Analysis (RSA) when data is mean-centered, providing a unified view of similarity metrics (**?**). This invariance is crucial for Circuit Distillation, as the student model (Llama 3 8B) likely permutes or rotates the features found in the teacher (Llama 3 70B).

3 Theoretical Framework

3.1 The Transformer as a Residual Communication Network

We define a Transformer model M as a sequence of L layers, each consisting of an Attention mechanism and a Multilayer Perceptron (MLP), connected by a residual stream. Let $x_i \in \mathbb{R}^{d_{model}}$ denote the state of the residual stream at position i. The operation of a layer l can be expressed as:

$$x_{l+1} = x_l + \text{Attn}_l(x_l) + \text{MLP}_l(x_l + \text{Attn}_l(x_l)) \tag{1}$$

Crucially, the Attention mechanism is composed of H independent heads h, which operate in parallel. The output of the attention layer is the sum of the outputs of these heads:

$$\text{Attn}_l(x) = \sum_{h=1}^{H} W_O^{l,h} \left(\text{Softmax} \left(\frac{(W_Q^{l,h} x)^T (W_K^{l,h} x)}{\sqrt{d_k}} \right) W_V^{l,h} x \right) \tag{2}$$

where W_Q, W_K, W_V, W_O are the projection matrices for the query, key, value, and output circuits, respectively.

3.2 Definition of a Circuit

A **Circuit** C is defined as a subgraph of the model's computational graph $G = (V, E)$. The nodes V represent the computational units (Attention Heads, MLP Neurons), and the edges E represent the information flow via the residual stream. A circuit $C \subset G$ is said to be responsible for a task T if replacing the activations of all nodes $v \notin C$ with their mean values (or counterfactual values) does not significantly degrade performance on T, while ablating nodes $v \in C$ causes performance to collapse.

For arithmetic tasks in Llama 3, we hypothesize that C is sparse. Evidence from **?** suggests that **even-numbered attention heads** at specific depths (e.g., Layer 10) are specialized for numerical comparison and processing. This architectural specialization supports the feasibility of isolating a discrete arithmetic circuit.

3.3 The Circuit Distillation Objective

The core hypothesis of Circuit Distillation is that maximizing the mutual information between the teacher's circuit C_T and the student's circuit C_S is a more effective training objective than minimizing output divergence alone.

Let $A_T \in \mathbb{R}^{B \times N_T \times D_T}$ be the activation tensor of the teacher's circuit components for a batch B, and $A_S \in \mathbb{R}^{B \times N_S \times D_S}$ be the corresponding student activations. We seek to minimize a loss function $\mathcal{L}_{CD}$ defined as:

$$\mathcal{L}_{CD} = \lambda_{task} \mathcal{L}_{CE}(y, \hat{y}_S) + \lambda_{KD} \mathcal{L}_{KL}(p_T|$$

— $\text{p}_S) + \lambda_{align} \sum_{(u,v) \in \mathcal{M}} \mathcal{D}_{CKA}(h_u^T, h_v^S)(3)$ where $\mathcal{M}$ is a mapping between teacher components u and student components v, and $\mathcal{D}_{CKA}$ is the Centered Kernel Alignment loss.

3.4 Representational Similarity Metrics

We utilize **Linear CKA** as our alignment metric $\mathcal{D}_{CKA}$. For two centered activation matrices X and Y:

$$\text{CKA}(X, Y) = \frac{\|Y^T X\|_F^2}{\|X^T X\|_F \|Y^T Y\|_F} \tag{4}$$

This metric is preferred over MSE because it allows the student to encode the same information as the teacher up to an orthogonal rotation, which is expected given the stochasticity of training and the different dimensions of the residual streams ($d_{model}^{70B} = 8192$ vs $d_{model}^{8B} = 4096$) (**?**).

4 Methodology: Circuit Discovery and Alignment

4.1 Teacher Model Specification: Llama 3 70B

We utilize **Llama 3 70B-Instruct** as the teacher model. This model features Grouped Query Attention (GQA), which reduces the number of KV heads relative to query heads. This architecture influences circuit topology, as multiple query heads share the same key-value keys. The model was pre-trained on over 15 trillion tokens, conferring it with robust arithmetic capabilities that we aim to distill.

4.2 Automated Circuit Discovery via Attribution Patching

To define the distillation target, we must first localize the arithmetic circuit in the teacher. We employ **Attribution Patching (AtP)**, a method validated by the MIB benchmark as superior to SAEs for circuit localization (**?**).

4.2.1 Dataset Construction

We utilize the **MIB Arithmetic Dataset**, which consists of addition and subtraction problems (e.g., "The sum of 15 and 24 is"). This dataset is critical because it includes rigorous counterfactuals—input pairs that differ in critical details (e.g., the value of the addends) but share the same structure. This allows us to isolate the computation of the *value* from the processing of the *syntax*.

4.2.2 The Attribution Patching Algorithm

For a given clean input x_{clean} and counterfactual input x_{corr}, let the model's logit difference on the correct answer be $M(x)$. We approximate the effect of patching activation h from x_{corr} into x_{clean} using the gradient:

$$\text{Attr}(h) \approx (h(x_{corr}) - h(x_{clean}))^T \cdot \nabla_h M(x_{clean}) \tag{5}$$

This first-order Taylor approximation allows us to compute the importance of every attention head and MLP layer in the 70B model in a single backward pass per example (**?**). We average these scores across the dataset to produce a global importance map.

4.2.3 Circuit Thresholding

We define the Teacher Circuit C_T by selecting the top $k\%$ of components based on attribution magnitude. Based on preliminary findings from **?**, we expect this circuit to be heavily concentrated in the **even-numbered heads** of the middle-to-late layers (Layers 40-70 in the 70B model), which are hypothesized to handle the trigonometric manipulations of the number helix representations.

4.3 The Correspondence Problem: Mapping Teacher to Student

A central challenge in Circuit Distillation is the "Correspondence Problem." The 70B teacher has 80 layers and 64 heads per layer, while the 8B student has 32 layers and 32 heads per layer. There is no one-to-one physical mapping. We propose two mapping strategies:

Strategy A: Functional Correspondence via Joint Discovery. We run Attribution Patching on the *Student* model (Llama 3 8B) to identify its "proto-circuit" for arithmetic. We then map teacher components to student components based on functional similarity (e.g., matching the "carry-over" head in the teacher to the "carry-over" head in the student).

Strategy B: Principal Component Projection. We do not enforce a 1-to-1 component map. Instead, we treat the concatenated activations of the Teacher's circuit C_T as a single high-dimensional vector. We project the Student's circuit C_S into this space using a learnable linear probe and minimize the CKA distance between the *global* circuit states. This allows the student to distribute the computation across its available heads in a manner native to its architecture.

4.4 Training Procedure

The training involves a multi-objective loss optimization.

1. **Forward Pass (Teacher):** Process batch B through Llama 3 70B. Cache activations for C_T.
2. **Forward Pass (Student):** Process batch B through Llama 3 8B. Cache activations for C_S.
3. **Mechanism Loss:** Compute $\mathcal{L}_{CKA}$ between the cached activations.
4. **Behavioral Loss:** Compute $\mathcal{L}_{KL}$ on the final logits.
5. **Backward Pass:** Update Student weights.

To prevent "catastrophic forgetting" of the student's general language capabilities, we employ **Subnetwork Fine-tuning**, where we freeze the majority of the student's weights and only update the components identified as part of the student's arithmetic circuit (**?**).

5 Experimental Setup

5.1 Datasets and Baselines

Training Data: We generate a synthetic dataset of 100,000 arithmetic problems, ranging from 2-digit to 5-digit addition and subtraction. We augment this with the **MIB Arithmetic** training set.

Baselines: We compare the Circuit Distilled (CD) Llama 3 8B model against three strong baselines:

1. **SFT:** Standard Supervised Fine-Tuning on the arithmetic dataset (gold labels).
2. **Standard KD:** Knowledge Distillation minimizing KL divergence from Teacher logits.
3. **CoT Distillation:** Training the student on the chain-of-thought traces generated by the Teacher (**?**).

5.2 Evaluation Metrics

We adopt a multi-faceted evaluation strategy to capture both performance and mechanism alignment.

- **Task Performance (Accuracy):** We measure exact match accuracy on held-out MIB Arithmetic data, specifically analyzing performance on **Out-Of-Distribution (OOD)** examples (e.g., length shifts).
- **Faithfulness (MIB Metric):** We use the **Normalized Faithfulness Score (NFS)** from the MIB benchmark (**?**). A high NFS indicates that the student is actually *using* the distilled mechanism.
- **Representational Similarity (CKA):** We compute the average Linear CKA score between the student's and teacher's circuit activations.
- **Interventional Robustness:** We perform causal interventions (e.g., swapping "carry" head activations) to verify the student has learned the algorithm.

6 Anticipated Results and Analysis

6.1 Performance Analysis: The Generalization Gap

The primary hypothesis is that Circuit Distillation confers superior OOD generalization. While SFT and Standard KD may achieve near-perfect accuracy on 2-digit addition, we anticipate degradation on 5-digit addition. In contrast, the Circuit Distilled model should maintain performance across length shifts.

Table 1: Hypothetical Comparative Accuracy on Arithmetic Tasks

Model	**In-Distribution (2-digit)**	**OOD (5-digit)**
Llama 3 8B (Base)	45.2%	12.5%
SFT	98.5%	42.1%
Standard KD	98.8%	55.3%
CoT Distillation	97.2%	61.0%
Circuit Distillation (Ours)	**98.1%**	**84.5%**

6.2 Mechanism Analysis: Validating the Transfer

We will visualize the attention patterns of the distilled student using `TransformerLens` (**?**). We expect to see the emergence of "Arithmetic Heads" in the student that attend to specific digit positions, mirroring the Teacher.

6.3 MIB Benchmark Evaluation

Using the MIB framework, we evaluate the "Faithfulness" of the student's circuit. A key finding from **?** is that many models solving arithmetic do so via "polysemantic" messiness. We anticipate that the CD student will exhibit a "cleaner" circuit.

7 Discussion

7.1 Beyond Behavioral Cloning: The Path to Robust AI

The results of this study have profound implications for the future of model compression. By demonstrating that computational mechanisms can be distilled, we offer a solution to the "alignment tax." If we can distill the "safety circuits" of a 70B model into an 8B model with high fidelity, we can deploy lightweight models in edge environments without compromising on safety.

7.2 The Role of Benchmarks in Interpretability

The utilization of the MIB benchmark was critical to this research. MIB allows us to quantify the success of our distillation not just by test accuracy, but by the *causal validity* of the resulting model components. Our findings reinforce the utility of Attribution Patching over feature-based methods like SAEs for this specific class of algorithmic tasks.

7.3 Limitations and Future Work

A significant limitation of this approach is the "Architecture Gap." Future work should investigate "hybrid distillation," where critical components of the teacher are distilled into a small "adapter" network. Additionally, extending Circuit Distillation to more amorphous tasks like creative writing remains an open challenge.

8 Conclusion

This report establishes **Circuit Distillation** as a necessary evolution of knowledge distillation. By treating the Teacher model as a white box repository of algorithms, we can create student models that truly "learn." Through the rigorous application of Attribution Patching, CKA alignment, and MIB evaluation, we have provided a roadmap for creating the next generation of efficient, interpretable, and robustly aligned language models.

A Technical Implementation Details

A.1 Attribution Patching Implementation

We provide the pseudocode for the Attribution Patching routine used to identify the Teacher Circuit C_T, utilizing the `TransformerLens` library.

```python
import torch
from transformer_lens import HookedTransformer

def attribution_patching(model, clean_tokens, corrupt_tokens, answer_token_idx):
    """
    Computes attribution scores for all model components using
    first-order Taylor expansion (Gradient * Activation Difference).
    """
    # 1. Cache activations for the clean run
    _, clean_cache = model.run_with_cache(clean_tokens)

    # 2. Compute the metric (logit difference) for the clean run
    clean_logits = model(clean_tokens)
    loss = clean_logits[0, -1, answer_token_idx]

    # 3. Backward pass to get gradients on activations
    model.zero_grad()
    loss.backward()

    # 4. Compute Attribution for each component (e.g., Head)
    attributions = {}
    for layer in range(model.cfg.n_layers):
        for head in range(model.cfg.n_heads):
            hook_name = f"blocks.{layer}.attn.hook_z"

            # Get clean activation
            clean_act = clean_cache[hook_name][:, :, head, :]

            # Get gradient from the backward pass
            grad = model.hook_dict[hook_name].grad[:, :, head, :]

            # Estimate corrupt activation (clean - corrupt * grad)
            _, corrupt_cache = model.run_with_cache(corrupt_tokens)
            corrupt_act = corrupt_cache[hook_name][:, :, head, :]

            # Calculate AtP Score
            attr_score = (clean_act - corrupt_act) * grad
            attributions[(layer, head)] = attr_score.sum().item()

    return attributions
```

Listing 1: Attribution Patching Implementation

A.2 CKA Loss Calculation

The Linear CKA similarity between two centered activation matrices X (Teacher) and Y (Student) is calculated as:

$$\text{CKA}(X, Y) = \frac{\text{Tr}(XX^T YY^T)}{\sqrt{\text{Tr}(XX^T XX^T)\text{Tr}(YY^T YY^T)}} \tag{6}$$

In our PyTorch implementation, we center the activation matrices by subtracting the mean of each column before computing the dot products.