DISTILLING MODEL-INHERENT REASONING CIRCUIT FOR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge Distillation (KD) is widely adopted to compress Large Language Models (LLMs) by aligning the output distributions of teacher model and student model. However, output matching alone does not explicitly characterize the internal computations reflecting the teacher's reasoning capability, and directly aligning the representations of paired Transformer components risks disrupting the student's intrinsic reasoning pattern. To address these limitations, we propose model-inherent Reasoning Circuit Distillation (RCD). RCD characterizes model-inherent reasoning patterns as critical circuits formed by Transformer edges, with a separate circuit identified for each model using Edge Attribution Patching with Integrated Gradients. Subsequently, to transfer the teacher's reasoning capability, RCD aligns the predictive effects of both circuits by matching the logit variations induced by circuit attenuation. Theoretically, we establish guarantees for recovering the ground-truth reasoning circuits and show that the empirical RCD loss controls the population discrepancy in predictive effects between the ground-truth circuits of the teacher model and the student model. Experiments on mathematical reasoning and code generation show that RCD consistently outperforms state-of-the-art KD methods.

1 INTRODUCTION

Scaling the number of parameters and the volume of training data has substantially improved the capabilities of Large Language Models (LLMs) on complex tasks (Kaplan et al., 2020). However, the increasing number of parameters makes LLMs difficult to deploy in scenarios with limited computational resources. Therefore, Knowledge Distillation (KD) (Hinton et al., 2015) is now widely adopted for LLM compression (Ko et al., 2025; On et al., 2026), which transfers knowledge from a large teacher model to a compact student model while preserving most of the teacher's capabilities.

Recently, several KD methods have been proposed to shrink the capability gap between teacher model and student model (Zhang et al., 2023; Li et al., 2025; 2026). They often attempt to align the output distributions or the intermediate feature representations of the teacher model and the student model. For example, to strike a balance between mode seeking and mode covering, DistiLLM-2 (Ko et al., 2025) simultaneously optimizes the skewed Kullback–Leibler (SKL) divergence on teacher-generated outputs and the skew reverse Kullback–Leibler (SRKL) divergence on student-generated outputs in an on-policy manner. Moreover, ABKD (Wang et al., 2025a) is proposed to distil knowledge from a fixed dataset by optimizing the α-β-divergence, which is a generalization of the SKL divergence and the SRKL divergence. Despite their effectiveness, a key challenge lies in characterizing the internal mechanisms carrying the teacher's reasoning capability and leveraging the resulting insights to guide the transfer of its reasoning capability to the student model.

Therefore, in this paper, we attempt to use the circuit-level reasoning patterns (Bhaskar et al., 2024) to characterize the internal computations within LLMs. Specifically, we find that teacher model and student model have learnt distinct circuit-level reasoning patterns during the pre-training phase. Specific examples of reasoning circuits for Qwen2.5-Coder-7B-Instruct and Qwen2.5-Coder-1.5B models are shown in Fig. 1, in which the two circuits characterize the sparse computational edges that are most important for code generation task. Here, the edges represent directed information flow between attention modules and MLP blocks. Since the two models have the same depth, it can be observed that the model with 1.5B parameters invokes more attention heads in the intermediate lay-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

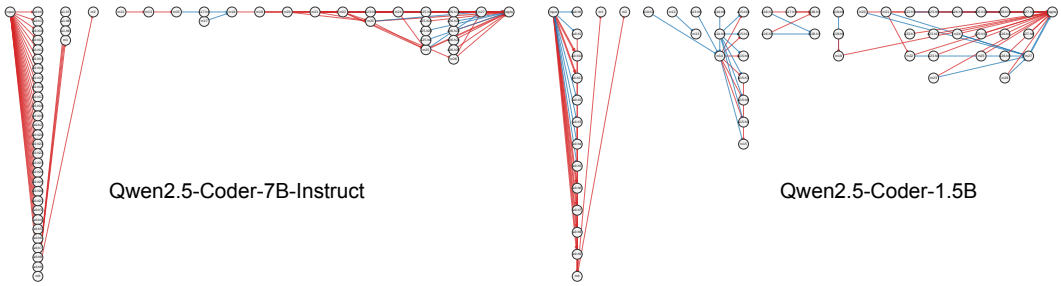


Figure 1: Reasoning circuits of two Qwen2.5-Coder models, each containing the 80 most critical edges for the code generation task. The blue and red edges indicate positive effects and negative effects, respectively, while thicker edges indicate greater significance. The sparser 7B circuit (left panel) suggests less intermediate computation than its 1.5B counterpart (right panel).

ers to extract and process the input information, while the model with 7B parameters has a simpler forward pattern. Therefore, we argue that models with different architectures or capacities have intrinsically different reasoning patterns (*a.k.a.* reasoning circuits) formed during pre-training. Prior studies on mechanistic interpretability also support this claim (Haskins & Adams, 2025; Gibson et al., 2026). Given these model-inherent circuits, directly aligning the representations of functionally corresponding attention heads or MLP blocks across models, as in Circuit Distillation (Wadhwa et al., 2025), may disrupt the student's intrinsic reasoning pattern.

To maintain the reasoning pattern learned during pre-training, we propose model-inherent Reasoning Circuit Distillation (RCD). To be specific, RCD identifies the critical circuit formed by Transformer edges for both teacher and student via the Edge Attribution Patching (EAP) technique. Given a specific task, EAP computes the attribution score of each edge, and the circuit containing edges with the highest scores characterizes the intrinsic reasoning patterns of a model. Since evaluating every edge through a separate intervention requires millions of forward passes, which is computationally prohibitive, an approximation method with integrated gradients (namely EAP-IG (Hanna et al., 2024)) is adopted. After the identification of model-inherent reasoning circuits, RCD matches the predictive effects of the two circuits by aligning the logit variations caused by attenuating each circuit. To capture different transitions in reasoning (Dutta et al., 2024; Cabannes et al., 2024), we discover circuits under three modes, namely, prompt-to-reasoning, step-to-next-step, and reasoning-to-answer. These modes capture how reasoning begins, progresses between steps, and produces the final answer, respectively. Overall, RCD aligns predictive effects through each model's own reasoning circuit to narrow the capacity gap between teacher and student.

Our contributions are summarized as follows:

- We find that different LLMs have distinct reasoning circuits (namely, sparse computational graphs formed by Transformer edges) learned during pre-training. Based on this observation, we propose RCD to align the effects induced by model-inherent reasoning circuits for teacher and student.
- Theoretically, we are the first to analyze the approximation error of the circuit discovered by EAP-IG relative to the ground-truth circuit defined by a faithfulness criterion. We also derive an upper bound on the discrepancy in predictive effects between the teacher's and student's ground-truth reasoning circuits, which provides a theoretical justification for RCD.
- Empirically, extensive experiments on mathematical reasoning task and code generation task demonstrate the superiority of our method over state-of-the-art knowledge distillation methods.

2 Related Work

We briefly review related works on mechanistic interpretability of LLMs and knowledge distillation.

2.1 Mechanistic Interpretability of LLMs

Mechanistic interpretability explains model behavior through the computations that causally produce it. Early studies manually reverse-engineer sparse circuits for behaviors such as indirect object

identification and numerical comparison (Wang et al., 2022; Hanna et al., 2023), while later work localizes chain-of-thought operations, long-context retrieval, and in-context task representations to task-relevant components (Dutta et al., 2024; Wu et al., 2025; Todd et al., 2024). To reduce the reliance on manual hypotheses, Automatic Circuit DisCovery (ACDC) greedily removes edges with small activation-patching effects (Conmy et al., 2023), while Edge Attribution Patching (EAP) uses a first-order approximation to score all edges efficiently (Syed et al., 2024).

Edge Attribution Patching with Integrated Gradients (EAP-IG) further integrates gradients along the counterfactual-to-clean path to reduce endpoint approximation error and recover more faithful circuits (Hanna et al., 2024). Nevertheless, a recovered circuit remains conditional on the model, task metric, and intervention (Zhang & Nanda, 2024), while structural overlap alone does not guarantee functional faithfulness. This has implications for distillation: a student may reproduce the behavior of the teacher model by reorganizing, compressing, or discarding internal components (Haskins & Adams, 2025). We therefore discover a separate model-inherent circuit for each model and align their predictive effects, rather than requiring their structures or individual components to coincide.

2.2 Knowledge Distillation

Traditional Knowledge Distillation (KD) transfers knowledge from a teacher model to a student model by matching output distributions (Zhao et al., 2022), intermediate representations (Chen et al., 2021b), or feature relations (Yang et al., 2022). For LLMs, KD enables substantially smaller models to acquire strong reasoning capabilities from powerful teachers, as shown by the models distilled from DeepSeek-R1 (Guo et al., 2025). Representative KD methods for LLMs include Generalized Knowledge Distillation (GKD) (Agarwal et al., 2024), which trains the student on self-generated sequences using teacher feedback, and DistiLLM-2 (Ko et al., 2025), which contrasts teacher- and student-generated outputs in an on-policy objective. These methods improve training distributions and divergence objectives but still characterize transferred knowledge through output probabilities.

Other methods address the capacity gap (Zhang et al., 2023; Li et al., 2025; 2026) through intermediate distributions or more flexible divergence objectives. Representative methods include Temporally Adaptive Interpolated Distillation (TAID) (Shing et al., 2025), ABKD (Wang et al., 2025a), and α-Mixture Assistant Distribution (AMiD) (Shin et al., 2026). Beyond logit distillation, Causal Distillation transfers causally relevant internal information (Wu et al., 2022), whereas Circuit Distillation matches functionally corresponding components and aligns their representations (Wadhwa et al., 2025). However, component correspondence may not hold across models (Haskins & Adams, 2025), and component-wise alignment may disrupt the student's intrinsic reasoning pattern.

3 Problem Definition

In this paper, we study Knowledge Distillation (KD) from a large teacher model $\mathcal{M}_{\mathrm{T}}(\cdot;\boldsymbol{\theta}_{\mathrm{T}})$ parameterized by $\boldsymbol{\theta}_{\mathrm{T}}$ to a compact student model $\mathcal{M}_{\mathrm{S}}(\cdot;\boldsymbol{\theta}_{\mathrm{S}})$ parameterized by $\boldsymbol{\theta}_{\mathrm{S}}$, with the goal of shrinking their capability gap (Ko et al., 2025). We consider two logic-intensive tasks, *i.e.*, mathematical reasoning and code generation. For concision, we use $M \in \{\mathrm{T},\mathrm{S}\}$ to denote either model. Let $\mathcal{D} = \{(x^i, y^i)\}_{i=1}^{N}$ denote the distillation data, where x^i is the prompt or query for the i-th example and $y^i = (y^i_1, \ldots, y^i_{T_i})$ is the corresponding response of length T_i. We also use $X^i = (x^i, y^i)$ to denote the i-th example and the random variable X to denote a generic task example. Moreover, we use $\mathbf{E}_M(X) \in \mathbb{R}^{|X|\times d_M}$ to denote the input embedding of X, where $|X|$ is the sequence length and d_M is the embedding dimension of $\mathcal{M}_M$. Under teacher forcing, the token-level distributions of $\mathcal{M}_{\mathrm{T}}$ and $\mathcal{M}_{\mathrm{S}}$ over a shared vocabulary $\mathcal{V}$ are $p^i_t(v) = p_{\mathcal{M}_{\mathrm{T}}}(v|x^i, y^i_{<t})$ and $q^i_{\boldsymbol{\theta}_{\mathrm{S}},t}(v) = p_{\mathcal{M}_{\mathrm{S}}}(v|x^i, y^i_{<t})$, $\forall\, v \in \mathcal{V}$, respectively.

The naive method (Hinton et al., 2015) for KD minimizes the forward Kullback–Leibler (KL) divergence between the prediction distributions of $\mathcal{M}_{\mathrm{T}}$ and $\mathcal{M}_{\mathrm{S}}$. The loss function is given by

$$\mathcal{L}_{\mathrm{KD}}(\boldsymbol{\theta}_{\mathrm{S}}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{T_i}\sum_{t=1}^{T_i} D_{\mathrm{KL}}\big(p^i_t \big\| q^i_{\boldsymbol{\theta}_{\mathrm{S}},t}\big)\,, \text{ where } D_{\mathrm{KL}}(p\|q) = \sum_{v\in\mathcal{V}} p(v)\log\frac{p(v)}{q(v)}. \tag{1}$$

In practice, some advanced loss functions can be adopted for KD, such as the α-β-divergence proposed in ABKD (Wang et al., 2025a) and the mixture of the forward KL divergence and the reverse KL divergence leveraged in DistiLLM-2 (Ko et al., 2025).

4 METHOD

Existing KD methods primarily transfer knowledge by matching the output distributions of $\mathcal{M}_{\mathrm{T}}$ and $\mathcal{M}_{\mathrm{S}}$, without explicitly characterizing the internal computations underlying their reasoning. In view of this, we propose to characterize the reasoning capability of a model via its circuit-level reasoning pattern (*a.k.a.*, reasoning circuit), which is formed by edges between Transformer components (*e.g.*, attention heads or MLP blocks). We find that models with different architectures or capacities may solve the same task through distinct reasoning circuits, as shown in Fig. 1 and Appendix A. Prior studies on mechanistic interpretability also support this claim (Haskins & Adams, 2025; Gibson et al., 2026). However, directly aligning the representations of attention heads or MLP blocks across these circuits may disrupt the intrinsic reasoning pattern of $\mathcal{M}_{\mathrm{S}}$ and degrade its performance.

To preserve the model-inherent reasoning pattern of student model while transferring task-relevant reasoning capability from teacher model, we propose Reasoning Circuit Distillation (RCD). RCD aligns the effects on prediction induced by the reasoning circuits of $\mathcal{M}_{\mathrm{T}}$ and $\mathcal{M}_{\mathrm{S}}$. Specifically, in Section 4.1, we independently discover faithful circuits for $\mathcal{M}_{\mathrm{T}}$ and $\mathcal{M}_{\mathrm{S}}$ by identifying critical computational edges for different transitions within each reasoning trace. In Section 4.2, we distill the effects of these model-inherent circuits on predictions at informative response positions.

4.1 FAITHFUL MODEL-INHERENT REASONING CIRCUIT DISCOVERY

Since computation in LLMs is distributed across Transformer edges, we first identify a faithful reasoning circuit (*i.e.*, a set of Transformer edges that contribute to task-relevant predictions) for each model. To be specific, for $M \in \{\mathrm{T}, \mathrm{S}\}$, we view $\mathcal{M}_M$ as a directed graph $\mathcal{G}_M = (\mathcal{N}_M, \mathcal{E}_M)$, where the nodes in set $\mathcal{N}_M$ are components such as attention heads and MLP blocks and the edges in set $\mathcal{E}_M$ transmit activations between them. The total number of edges is $E_M = |\mathcal{E}_M|$. Let X be a task example and X^{cf} be a counterfactual example sampled independently from a given dataset $\mathcal{P}$, which is disjoint from $\mathcal{D}$. We use the log-likelihood $\mathrm{LL}_M(X) := {}^1\!/_{|\mathcal{T}(X)|} \sum_{t \in \mathcal{T}(X)} \log p_{\mathcal{M}_M}(y_t \mid X_{<t})$ as the objective in circuit discovery, where $\mathcal{T}(X)$ denotes the index set for response tokens.

In the following, we adopt Edge Attribution Patching with Integrated Gradients (EAP-IG) (Hanna et al., 2024) to identify the reasoning circuit for a given task. Here, the original edge attribution patching method applies causal interventions (Vig et al., 2020) to edge activations and attributes the resulting changes in task predictions to individual edges. For an edge $e = (u_e, v_e)$ with source node u_e and target node v_e and an example X, let $\boldsymbol{z}^{\mathrm{out}}_{M,u_e}(X) \in \mathbb{R}^{|X| \times d_M}$ be the source activation of e. The attribution score of $e \in \mathcal{E}_M$ defined by EAP is

$$s_{M,e} := \mathbb{E}_{X,X^{\mathrm{cf}}}[\mathrm{LL}_M(X) - \mathrm{LL}_M(X; \mathrm{do}(e \leftarrow \boldsymbol{z}^{\mathrm{out}}_{M,u_e}(X^{\mathrm{cf}})))], \tag{2}$$

where $\mathrm{do}(\cdot)$ denotes an intervention (*a.k.a.*, patching) that replaces the edge activation with its counterfactual value (Syed et al., 2024). Since it is computationally prohibitive to calculate $s_{M,e}$ for all edges in an LLM, EAP-IG approximates it by interpolating between the two input embeddings as $\mathbf{E}^{(\alpha)}_M(X, X^{\mathrm{cf}}) = (1-\alpha)\mathbf{E}_M(X^{\mathrm{cf}}) + \alpha \mathbf{E}_M(X) \in \mathbb{R}^{|X| \times d_M}$ for $\alpha \in [0,1]$. It then calculates estimated scores with this shared input. Given n pairs of examples $\{(X_i, X_i^{\mathrm{cf}})\}_{i=1}^n$ and, for each pair, m interpolation points $\{\mathbf{E}^{(j/m)}_M(X_i, X_i^{\mathrm{cf}})\}_{j=1}^m$, the estimated attribution score for $e \in \mathcal{E}_M$ is

$$\widehat{s}_{M,e} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left\langle \boldsymbol{d}_{M,e}(X_i, X_i^{\mathrm{cf}}), \nabla_{\boldsymbol{z}^{\mathrm{in}}_{M,v_e}} \mathrm{LL}^{(j/m)}_{M,i} \right\rangle, \tag{3}$$

where $\boldsymbol{d}_{M,e}(X, X^{\mathrm{cf}}) = \boldsymbol{z}^{\mathrm{out}}_{M,u_e}(X) - \boldsymbol{z}^{\mathrm{out}}_{M,u_e}(X^{\mathrm{cf}}) \in \mathbb{R}^{|X| \times d_M}$ is the activation difference, $\mathrm{LL}^{(\alpha)}_{M,i}$ is the log-likelihood for input embedding $\mathbf{E}^{(\alpha)}_M(X_i, X_i^{\mathrm{cf}})$, and $\boldsymbol{z}^{\mathrm{in}}_{M,v_e}$ is the input of node v_e. Based on the scores, we can construct a circuit with K_{eg} edges by selecting the edges with the largest $|\widehat{s}_{M,e}|$.

However, prior mechanistic studies indicate that different stages of information processing can rely on different Transformer components. In particular, early MLP blocks and attention heads primarily enrich and propagate relevant information, while later induction heads directly promote output tokens (Dutta et al., 2024; Geva et al., 2023; Wang et al., 2022). Motivated by these findings, we extend the above circuit discovery procedure to account for three transitions within the reasoning process of each task example. We refer to these transitions as modes and denote the set of modes by

$\mathcal{R} = \{\text{prompt-to-reasoning}, \text{step-to-next-step}, \text{reasoning-to-answer}\}$. Here, "prompt-to-reasoning" uses the prompt tokens and reasoning tokens as the source span and target span, respectively. Under the same source-to-target convention, "step-to-next-step" takes an intermediate reasoning step as its source span and the immediately following step as the target span, whereas "reasoning-to-answer" takes the complete reasoning trace as the source span and the final answer as the corresponding target span. To identify the edges that causally transmit information from the source span to the target span, the EAP-IG method is applied separately to each mode. For a given mode, the intervention $\mathrm{do}(\cdot)$ replaces the edge activations at the tokens in the source span of X with the corresponding activations from a length-matched span in X^{cf}, and $\mathrm{LL}_M(X)$ is evaluated only over the tokens in the target span. This mode-specific attribution characterizes edge roles in reasoning initiation, stepwise progression, and answer formation, as illustrated by a practical example in Appendix B.

Consequently, for a specific mode r, we use $s^r_{M,e}$ and $\widehat{s}^r_{M,e}$ to represent the ground-truth attribution score and the score estimated via Eq. (3), respectively. The circuit $\widehat{C}^r_M$ for mode r is then derived by selecting K_{eg} edges in $\mathcal{E}_M$ with the largest $|\widehat{s}^r_{M,e}|$, and the circuit leveraged for distillation below is the union of $\widehat{C}^r_M$ over all $r \in \mathcal{R}$. Although the discovered circuit for each mode contains exactly K_{eg} edges, the overlaps among these circuits may differ across models. Consequently, the independently discovered union circuits $\widehat{C}_{\mathrm{T}}$ and $\widehat{C}_{\mathrm{S}}$ may contain different numbers and types of edges.

After circuit discovery, we further characterize the functional sufficiency of the discovered circuit relative to the intact LLM using the faithfulness metric (Méloux et al., 2026). For a circuit C, we retain the edges in C and patch every edge outside C with its activation from X^{cf}, which yields the population log-likelihood $\overline{\mathrm{LL}}_M(C) := \mathbb{E}_X[\mathrm{LL}_M(X; C)]$. The faithfulness of C is then defined as

$$\mathrm{Faith}_M(C) = \frac{\overline{\mathrm{LL}}_M(C) - \overline{\mathrm{LL}}_M(\varnothing)}{\overline{\mathrm{LL}}_M(\mathcal{E}_M) - \overline{\mathrm{LL}}_M(\varnothing)}. \tag{4}$$

Therefore, $C = \varnothing$ and $C = \mathcal{E}_M$ correspond to fully patched model and intact model, respectively. The faithfulness metric is then used in Sec. 5 to define the ground-truth reasoning circuit of an LLM.

4.2 Reasoning Circuit Distillation

Based on the discovered circuits of teacher and student, we propose Reasoning Circuit Distillation (RCD) to align their effects on prediction. This design aims to preserve the inherent reasoning pattern learned by $\mathcal{M}_{\mathrm{S}}$ during pre-training, which is critical for performance improvement of student model. The proposed RCD comprises three operations: obtaining clean predictions from $\mathcal{M}_{\mathrm{T}}$ and $\mathcal{M}_{\mathrm{S}}$, attenuating the model-inherent circuit in each model to measure the induced logit variation, and aligning the two variations at selected tokens. In detail, for a response position t, let $\boldsymbol{z}_{M,t}(C; X^i, \rho) \in \mathbb{R}^{|\mathcal{V}|}$ denote the logit vector of $\mathcal{M}_M$ over vocabulary $\mathcal{V}$ when the activation transmitted by every edge in C is scaled by $1-\rho$. Here, $\rho \in [0, 1]$ is a hyperparameter that controls edge attenuation. The effect on token prediction induced by the discovered circuit $\widehat{C}_M$ is defined as

$$\boldsymbol{\Delta}_{M,t}(\widehat{C}_M; X^i, \rho) = \boldsymbol{z}_{M,t}(\varnothing; X^i, \rho) - \boldsymbol{z}_{M,t}(\widehat{C}_M; X^i, \rho), \quad \forall\, M \in \{\mathrm{T}, \mathrm{S}\}. \tag{5}$$

This joint intervention captures the logit variation caused by attenuating all edges in $\widehat{C}_M$.

Next, it remains to determine the response positions t for effect alignment. It has been found that response tokens with high predictive entropy are particularly informative because they are critical forks that can steer the model toward different reasoning paths (Wang et al., 2025b). Therefore, distilling prediction effects at these tokens transfers the task-relevant reasoning capability of $\mathcal{M}_{\mathrm{T}}$. Accordingly, we perform effect alignment at tokens with the largest predictive entropy. Given a mini-batch $\mathcal{B} = \{X^i\}_{i=1}^{|\mathcal{B}|} \subset \mathcal{D}$ and the token budget K_H, we define the entropy at position t for the i-th example as $H^i_t = -\sum_{v \in \mathcal{V}} p^i_t(v) \log p^i_t(v)$. Then, we allocate this budget across responses in proportion to their lengths, and the selected set for X^i is denoted by $\mathcal{H}^i$. Subsequently, we align the prediction effects of the two model-inherent circuits $\widehat{C}_{\mathrm{T}}$ and $\widehat{C}_{\mathrm{S}}$ using

$$\mathcal{L}_{\mathrm{RCD}}(\boldsymbol{\theta}_{\mathrm{S}}; \rho) = \frac{1}{|\mathcal{B}|} \sum_{X^i \in \mathcal{B}} \frac{1}{|\mathcal{H}^i|} \sum_{t \in \mathcal{H}^i} \left\| \mathrm{sg}\Big(\boldsymbol{\Delta}_{\mathrm{T},t}(\widehat{C}_{\mathrm{T}}; X^i, \rho)\Big) - \boldsymbol{\Delta}_{\mathrm{S},t}(\widehat{C}_{\mathrm{S}}; X^i, \rho) \right\|_2, \tag{6}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation. In this objective, $\mathcal{M}_{\mathrm{T}}$ provides a fixed target effect, and only $\boldsymbol{\theta}_{\mathrm{S}}$ is optimized. Unlike direct logit matching (Boizard et al., 2025), RCD aligns circuit effects via Eq. (6) without requiring correspondence between Transformer components.

Finally, RCD can be added as an auxiliary loss to any base distillation objective $\mathcal{L}_{\text{base}}(\boldsymbol{\theta}_{\text{S}})$ such as KL divergence and α-β-divergence (Wang et al., 2025a), and the complete training objective is

$$\mathcal{L}(\boldsymbol{\theta}_{\text{S}}) = \mathcal{L}_{\text{base}}(\boldsymbol{\theta}_{\text{S}}) + \lambda_{\text{RCD}}\mathcal{L}_{\text{RCD}}(\boldsymbol{\theta}_{\text{S}}; \rho), \tag{7}$$

where λ_{RCD} controls the strength of effect alignment between circuits. Due to space limitations, the complete algorithm description is deferred to Appendix B.

5 Theoretical Analyses

RCD minimizes the effect discrepancy between circuits discovered by EAP-IG with a finite sample, whereas its target is the population effect mismatch between the ground-truth reasoning circuits of $\mathcal{M}_{\text{T}}$ and $\mathcal{M}_{\text{S}}$. We bridge this gap by first establishing the circuit recovery theorem and then providing the upper bound for the population risk. All proofs in this section are deferred to Appendix C. In the following, we first define the ground-truth reasoning circuit of a given model in Definition 1.

Definition 1 (Ground-Truth Reasoning Circuit). For $0 \leq \epsilon < 1$, the ground-truth reasoning circuit of a model $\mathcal{M}_M, M \in \{\text{T}, \text{S}\}$ is the unique smallest circuit that preserves at least a $1-\epsilon$ fraction of the task log-likelihood, namely, $C_M^\star := \arg\min_{\varnothing \subsetneq C \subsetneq \mathcal{E}_M} \{|C| : \text{Faith}_M(C) \geq 1-\epsilon\}$.

This definition specifies a faithful circuit formed by a minimal number of Transformer edges. It remains to show when the circuit $\widehat{C}_M$ discovered by the EAP-IG method identifies this target. To this end, we consider Eq. (3) instantiated for mode r, with LL_M evaluated on the corresponding target span. For $e = (u_e, v_e)$ and a discovery pair (X, X^{cf}), we define the exact single-edge path as $\boldsymbol{z}_{M,u_e}^{\text{out},(\alpha)} := (1-\alpha)\boldsymbol{z}_{M,u_e}^{\text{out}}(X^{\text{cf}}) + \alpha \boldsymbol{z}_{M,u_e}^{\text{out}}(X), \alpha \in [0,1]$. Let $\boldsymbol{g}_{M,e}^r(\alpha)$ and $\widetilde{\boldsymbol{g}}_{M,e}^r(\alpha)$ denote the gradients of LL_M with respect to $\boldsymbol{z}_{M,v_e}^{\text{in}}$ along the exact single-edge path and the shared path with input embedding $\mathbf{E}_M^{(\alpha)}(X, X^{\text{cf}})$, respectively. Subsequently, we still require the following assumptions.

Assumption 1 (Boundedness). There exist $\overline{Z}_M, G_M, H_M, \kappa_M > 0$ such that $\forall\, r \in \mathcal{R}$, $e \in \mathcal{E}_M$, $C \subseteq \mathcal{E}_M$, and $\alpha, \alpha' \in [0,1]$, $\max\{\|\boldsymbol{z}_{M,u_e}^{\text{out}}(\cdot)\|_2, \|\boldsymbol{z}_{M,t}(C;\cdot,\rho)\|_2\} \leq \overline{Z}_M$, $\max\{\|\boldsymbol{g}_{M,e}^r(\alpha)\|_2, \|\widetilde{\boldsymbol{g}}_{M,e}^r(\alpha)\|_2\} \leq G_M$, $\|\widetilde{\boldsymbol{g}}_{M,e}^r(\alpha) - \boldsymbol{g}_{M,e}^r(\alpha)\|_2 \leq \kappa_M$, and $\|\widetilde{\boldsymbol{g}}_{M,e}^r(\alpha) - \widetilde{\boldsymbol{g}}_{M,e}^r(\alpha')\|_2 \leq H_M|\alpha - \alpha'|$.

Assumption 2 (Identifiability of Ground-truth Circuit via EAP). For $M \in \{\text{T}, \text{S}\}$ and a mode $r \in \mathcal{R}$, define the boundary margin of C_M^r as $\gamma_{M,r} := \min_{e \in C_M^r} |s_{M,e}^r| - \max_{e \notin C_M^r} |s_{M,e}^r|$. We assume $\gamma_{M,r} > 0$ and that the union of the circuits discovered by EAP for all the modes equals the ground-truth circuit in Definition 1, namely $C_M^\star = \bigcup_{r \in \mathcal{R}} C_M^r$.

Here, Assumption 2 requires that the faithful reasoning circuit can be discovered by the EAP method, which calculates the attribution score for each edge via Eq. (2). Based on the two assumptions, we derive the guarantee of exact circuit recovery in Theorem 1.

Theorem 1 (Exact Circuit Recovery of EAP-IG). *Under Assumptions 1 and 2, for $M \in \{\text{T}, \text{S}\}$, we further assume that the margin $\gamma_{M,r}$ satisfies $\underline{\gamma}_M := \min_{r \in \mathcal{R}}(\gamma_{M,r} - 4\overline{Z}_M\kappa_M) > 0$. If $m > 4\overline{Z}_M H_M/\underline{\gamma}_M$ and $n > 128\overline{Z}_M^2 G_M^2 \log(2E_M|\mathcal{R}|/\delta)/\underline{\gamma}_M^2$, then $\Pr(\widehat{C}_M = C_M^\star) \geq 1-\delta$.*

Theorem 1 ensures that EAP-IG recovers C_M^* with high probability when every mode has sufficient score separation and the number of discovery examples n and the number of interpolation points m are sufficiently large. Based on this theorem, we next show in Theorem 2 that the empirical effect alignment over the discovered circuit via EAP-IG generalizes to the population risk defined over circuit C_M^*, namely $R_{\text{GT}}(\boldsymbol{\theta}_{\text{S}}; \rho) = \mathbb{E}_X\left[1/|\mathcal{H}(X)| \sum_{t \in \mathcal{H}(X)} \|\boldsymbol{\Delta}_{\text{T},t}(C_{\text{T}}^\star; X, \rho) - \boldsymbol{\Delta}_{\text{S},t}(C_{\text{S}}^\star; X, \rho)\|_2\right]$. Here, $\mathcal{H}(X)$ denotes the index set of high-entropy response tokens selected for X according to the predictive entropy of $\mathcal{M}_{\text{T}}$, analogously to $\mathcal{H}^i$ in Eq. (6).

Theorem 2 (Effect Alignment of Ground-truth Circuits). *Under Assumptions 1 and 2, for each $\mathcal{M} \in \{\mathcal{M}_{\text{T}}, \mathcal{M}_{\text{S}}\}$ and $r \in \mathcal{R}$, let $a_{M,r} := \gamma_{M,r}/2 - 2\overline{Z}_M\kappa_M - \overline{Z}_M H_M/m$. Suppose $a_{M,r} > 0, \forall\, r \in \mathcal{R}$. Then, for every $\delta \in (0,1)$, with probability at least $1-\delta$, it holds that*

$$R_{\text{GT}}(\boldsymbol{\theta}_{\text{S}}; \rho) \leq \mathcal{L}_{\text{RCD}}(\boldsymbol{\theta}_{\text{S}}; \rho) + 2(\overline{Z}_{\text{T}} + \overline{Z}_{\text{S}})\sqrt{\frac{\log(2/\delta)}{2N}} + \frac{4}{\delta}\left(\overline{Z}_{\text{T}} p_{\text{T}} + \overline{Z}_{\text{S}} p_{\text{S}}\right), \tag{8}$$

where, for each $M \in \{\text{T}, \text{S}\}$, $p_M := \sum_{r \in \mathcal{R}} 2E_M \exp[-n a_{M,r}^2/(8\overline{Z}_M^2 G_M^2)]$ is an upper bound on the probability that the discovered circuit for $\mathcal{M}_M$ differs from $C_M^\star$.

Table 1: Comparison of pass@1 accuracy (%) for mathematical reasoning task. Results are reported as mean ± std over three independent trials, and the best KD result in each column is shown in **bold**.

Method	Qwen2-Math-7B-Instruct → Qwen2-Math-1.5B			Qwen2.5-Math-7B-Instruct → Qwen2.5-Math-1.5B		
	GSM8K	MATH	Avg.	GSM8K	MATH	Avg.
$\mathcal{M}_{\mathrm{T}}$	87.95	70.70	79.32	94.47	77.66	86.06
$\mathcal{M}_{\mathrm{S}}$	31.69	20.48	26.09	36.69	31.74	34.21
SFT	78.17	45.38	61.77	80.67	48.14	64.40
Naive KD (Hinton et al., 2015)	62.98 ± 4.75	36.14 ± 2.33	49.56 ± 3.38	77.33 ± 2.79	58.02 ± 4.32	67.68 ± 3.50
SeqKD (Kim & Rush, 2016)	75.84 ± 0.50	46.07 ± 0.40	60.96 ± 0.06	84.54 ± 0.69	69.17 ± 0.19	76.85 ± 0.36
ABKD (Wang et al., 2025a)	76.57 ± 0.53	53.74 ± 1.07	65.16 ± 0.80	84.28 ± 0.39	69.51 ± 0.48	76.90 ± 0.06
ResidualKD (On et al., 2026)	74.68 ± 0.99	40.11 ± 3.25	57.39 ± 1.15	75.79 ± 0.27	45.31 ± 1.40	60.55 ± 0.64
Causal Distillation (Wu et al., 2022)	69.90 ± 0.40	42.30 ± 0.95	56.10 ± 0.67	82.74 ± 1.77	66.16 ± 1.03	74.45 ± 1.35
Circuit Distillation (Wadhwa et al., 2025)	57.75 ± 3.41	29.77 ± 3.39	43.76 ± 3.21	68.46 ± 3.07	50.61 ± 1.43	59.54 ± 2.24
CSM (Kim et al., 2026)	68.39 ± 0.65	43.04 ± 1.45	55.71 ± 1.04	79.91 ± 2.22	63.90 ± 1.34	71.90 ± 1.65
AMiD (Shin et al., 2026)	75.79 ± 0.23	**54.55 ± 0.95**	65.17 ± 0.55	82.29 ± 0.76	68.57 ± 0.28	75.43 ± 0.46
ULD (Boizard et al., 2025)	78.09 ± 0.61	47.88 ± 0.11	62.98 ± 0.26	79.81 ± 0.50	52.67 ± 0.45	66.24 ± 0.38
Naive KD + RCD	69.14 ± 0.65	41.98 ± 0.37	55.56 ± 0.94	83.17 ± 0.40	67.80 ± 0.60	75.48 ± 0.48
ABKD + RCD	**78.51 ± 1.13**	54.14 ± 0.40	**66.32 ± 0.36**	**84.99 ± 0.69**	**69.52 ± 0.44**	**77.26 ± 0.33**

Theorem 2 decomposes the population risk with ground-truth circuits into the proposed RCD loss in Eq. (6), an $O(N^{-1/2})$ generalization gap, and a recovery term that decays exponentially with the number of discovery examples n. Therefore, the predictive effects of the ground-truth circuits can be aligned when RCD loss is small and both the discovery and alignment sample sizes are large.

6 Experiments

In this section, we evaluate the proposed RCD method on mathematical reasoning and code generation tasks and present ablation studies and further analyses.

6.1 Comparison with Existing KD Methods

Experimental setup. Following the experimental setup of DistiLLM-2 (Ko et al., 2025), for mathematical reasoning, we conduct distillation on 50K randomly selected query–response pairs from MetaMathQA (Yu et al., 2024) dataset. We adopt Qwen2-Math-7B-Instruct and Qwen2.5-Math-7B-Instruct as teacher models and Qwen2-Math-1.5B and Qwen2.5-Math-1.5B (Yang et al., 2024) as student models, respectively. For code generation, we construct training data from the Wizard-Coder corpus (Luo et al., 2024), which is developed using the Evol-Instruct method (Xu et al., 2024). We use Qwen2.5-Coder-7B-Instruct and DeepSeek-Coder-6.7B-Instruct as teacher models and Qwen2.5-Coder-1.5B and DeepSeek-Coder-1.3B (Hui et al., 2024) as student models, respectively. Additional details regarding the datasets, training configurations, and evaluation procedures are provided in Appendix D. For the proposed RCD method, we adopt the Naive KD loss function defined in Eq. (1) and the ABKD loss function (Wang et al., 2025a) as base loss functions.

Baseline methods. We compare RCD with representative KD methods. Specifically, the supervision-based baseline methods include Supervised Fine-Tuning (SFT), naive KD (Hinton et al., 2015), and Sequence-Level Knowledge Distillation (SeqKD) (Kim & Rush, 2016), which respectively learn from ground-truth responses, fixed token-level distributions of $\mathcal{M}_{\mathrm{T}}$, and sequences generated by $\mathcal{M}_{\mathrm{T}}$. The output-distribution-based KD methods include α-β Divergence (ABKD) (Wang et al., 2025a), Concrete Score Matching (CSM) (Kim et al., 2026), α-Mixture Assistant Distribution (AMiD) (Shin et al., 2026), and Universal Logit Distillation (ULD) (Boizard et al., 2025). We further compare with ResidualKD (On et al., 2026), which learns residual representations between $\mathcal{M}_{\mathrm{T}}$ and $\mathcal{M}_{\mathrm{S}}$; Causal Distillation (Wu et al., 2022), which aligns internal representations through interchange interventions; and Circuit Distillation (Wadhwa et al., 2025), which aligns functionally corresponding Transformer components, such as attention heads and MLP blocks.

The results in Tab. 1 and Tab. 2 demonstrate the superior performance of RCD across tasks and model families. For mathematical reasoning task, Naive KD + RCD improves the average accuracy of Naive KD by 6.00% and 7.80% on Qwen2-Math and Qwen2.5-Math, respectively. Additionally, ABKD + RCD achieves the highest average accuracies among KD methods. In particular, it exceeds

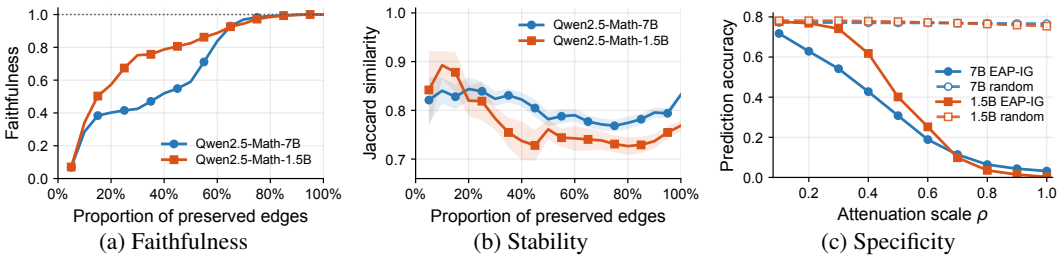
Table 2: Comparison of pass@1 accuracy (%) for code generation task. Results are reported as mean ± std over three independent trials, and the best KD result in each column is shown in **bold**.

Method	Qwen2.5-Coder-7B-Instruct → Qwen2.5-Coder-1.5B				
	HumanEval	HumanEval+	MBPP	MBPP+	Avg.
$\mathcal{M}_{\mathrm{T}}$	91.46	84.15	82.28	71.16	82.26
$\mathcal{M}_{\mathrm{S}}$	40.85	33.54	71.43	60.05	51.47
SFT	54.88	48.17	75.93	61.38	60.09
Naive KD (Hinton et al., 2015)	53.66 ± 3.17	45.93 ± 2.88	75.04 ± 0.77	63.05 ± 0.16	59.42 ± 1.28
SeqKD (Kim & Rush, 2016)	54.68 ± 0.70	47.76 ± 1.27	73.10 ± 1.25	60.40 ± 1.36	58.99 ± 1.09
ABKD (Wang et al., 2025a)	53.46 ± 1.27	45.73 ± 1.22	74.07 ± 0.53	62.87 ± 0.40	59.03 ± 0.48
ResidualKD (On et al., 2026)	46.75 ± 1.86	39.43 ± 1.86	68.25 ± 1.06	55.64 ± 1.25	52.52 ± 1.11
Causal Distillation (Wu et al., 2022)	56.51 ± 1.27	**49.19 ± 1.27**	74.07 ± 0.00	62.34 ± 0.55	60.53 ± 0.75
Circuit Distillation (Wadhwa et al., 2025)	38.01 ± 1.41	33.33 ± 0.93	61.38 ± 2.68	50.79 ± 2.52	45.88 ± 1.78
CSM (Kim et al., 2026)	52.85 ± 3.07	45.53 ± 3.92	74.69 ± 0.81	63.32 ± 0.85	59.10 ± 1.93
AMiD (Shin et al., 2026)	53.25 ± 2.31	45.32 ± 1.27	73.63 ± 0.76	62.61 ± 0.31	58.71 ± 0.90
ULD (Boizard et al., 2025)	51.83 ± 2.66	44.51 ± 2.80	73.46 ± 1.25	61.82 ± 1.88	57.90 ± 1.97
Naive KD + RCD	**56.71 ± 0.86**	48.14 ± 2.59	**75.40 ± 0.56**	**63.10 ± 0.19**	**60.84 ± 0.96**
ABKD + RCD	54.57 ± 1.29	47.26 ± 0.43	73.02 ± 0.37	62.43 ± 0.05	59.32 ± 0.52
Method	**DeepSeek-Coder-6.7B-Instruct → DeepSeek-Coder-1.3B**				
	HumanEval	HumanEval+	MBPP	MBPP+	Avg.
$\mathcal{M}_{\mathrm{T}}$	71.34	62.20	77.78	67.46	69.69
$\mathcal{M}_{\mathrm{S}}$	32.93	26.83	57.41	48.15	41.33
SFT	39.63	35.98	63.76	50.79	47.54
Naive KD (Hinton et al., 2015)	35.57 ± 0.35	31.10 ± 0.00	61.02 ± 0.15	50.00 ± 0.70	44.42 ± 0.14
SeqKD (Kim & Rush, 2016)	33.74 ± 1.27	28.46 ± 2.14	60.67 ± 0.40	49.65 ± 0.85	43.13 ± 0.91
ABKD (Wang et al., 2025a)	37.40 ± 1.27	32.32 ± 0.61	62.17 ± 1.06	50.97 ± 1.55	45.71 ± 1.05
ResidualKD (On et al., 2026)	36.39 ± 1.41	**32.93 ± 1.22**	61.46 ± 1.33	49.73 ± 1.15	45.13 ± 0.82
Causal Distillation (Wu et al., 2022)	35.78 ± 1.27	31.51 ± 1.27	61.38 ± 0.80	50.62 ± 0.80	44.82 ± 0.84
Circuit Distillation (Wadhwa et al., 2025)	27.13 ± 1.06	23.63 ± 1.04	15.81 ± 2.92	13.62 ± 2.35	20.05 ± 1.20
CSM (Kim et al., 2026)	35.98 ± 0.61	30.90 ± 0.35	61.20 ± 1.22	50.26 ± 1.59	44.58 ± 0.61
AMiD (Shin et al., 2026)	36.39 ± 0.35	31.10 ± 0.61	63.23 ± 1.47	51.68 ± 1.98	45.59 ± 1.03
ULD (Boizard et al., 2025)	36.39 ± 0.35	32.32 ± 0.61	62.88 ± 0.31	51.41 ± 0.31	45.75 ± 0.11
Naive KD + RCD	35.98 ± 0.93	30.49 ± 0.70	61.90 ± 0.79	49.74 ± 0.40	44.53 ± 0.37
ABKD + RCD	**39.63 ± 0.67**	**32.93 ± 0.46**	**63.76 ± 0.65**	**52.12 ± 0.55**	**47.11 ± 0.57**

Causal Distillation by 10.22% and 2.81% and Circuit Distillation by 22.56% and 17.72% in average accuracy. This indicates that directly enforcing representational alignment between paired Transformer components limits the performance improvement of student when teacher and student rely on different reasoning circuits. For code generation, Naive KD + RCD and ABKD + RCD achieve the highest average accuracies of 60.84% and 47.11% on Qwen2.5-Coder and DeepSeek-Coder, respectively. Moreover, when DeepSeek-Coder-6.7B-Instruct is adopted as the teacher model, our RCD outperforms AMiD by 1.52%, which suggests that RCD is more effective at alleviating the capacity gap problem (Li et al., 2026). It also outperforms logit-based distillation methods, such as CSM and ULD. Therefore, KD based on mechanistic interpretability is a promising direction.

6.2 Additional Ablation Studies

Analysis of discovered circuits. We analyze the faithfulness, stability, and specificity of the discovered circuits for the Qwen2.5-Math series, and the empirical results are illustrated in Fig. 2. Specifically, faithfulness is calculated via Eq. (4), which characterizes the task performance of the circuit constructed by preserving a certain proportion of edges with the largest estimated attribution scores $\widehat{s}_{M,e}$. Moreover, stability is evaluated via the Jaccard similarity (namely, $|C_1 \cap C_2|/|C_1 \cup C_2|$ for two circuits C_1 and C_2) across ten independent runs of the circuit discovery process, and specificity is evaluated by comparing the task performance of the discovered circuit with that of a size-matched circuit containing randomly selected Transformer edges. Fig. 2(a) shows that preserving the top 25% of edges ranked by $\widehat{s}_{M,e}$ retains 70% of the task capability of Qwen2.5-Math-1.5B. This finding is consistent with prior multi-task evaluations, which have demonstrated that EAP-IG is capable of identifying faithful circuits across a diverse range of capabilities (Hanna et al., 2024; Mueller et al., 2025). Additionally, Fig. 2(b) shows that the circuit estimates exhibit little variation across independent runs, providing evidence of statistical reliability that faithfulness alone cannot establish (Méloux et al., 2026). Finally, Fig. 2(c) shows that attenuating the discovered circuit degrades prediction accuracy far more than attenuating a size-matched circuit with randomly selected edges, which confirms that the discovered circuits are indispensable for accurate token prediction in LLMs.



(a) Faithfulness (b) Stability (c) Specificity

Figure 2: Analysis of discovered reasoning circuits for Qwen2.5-Math-7B-Instruct and Qwen2.5-Math-1.5B. (a) Faithfulness is calculated via Eq. (4); (b) stability is measured by pairwise Jaccard similarity across ten independent runs; and (c) specificity is assessed by comparing prediction accuracy after attenuating either the edges in the discovered circuit or randomly selected edges.

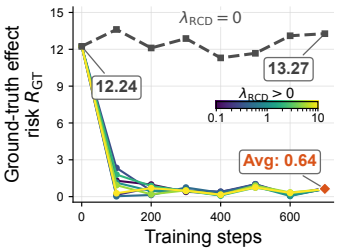


Figure 3: Effect alignment risk during training.

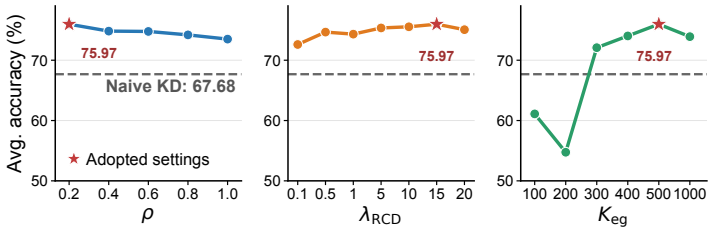


Figure 4: Sensitivity of Naive KD + RCD to ρ, λ_{RCD}, and K_{eg} on the Qwen2.5-Math series. The dashed line denotes the Naive KD method, and the red stars indicate the adopted settings.

Justification of theoretical results. To empirically examine Theorem 2, we evaluate the effect alignment risk of discovered circuits on a held-out dataset. This alignment risk is a surrogate for the ground-truth alignment risk R_{GT} in Eq. (8). Fig. 3 presents the risk dynamics during training and reveals that RCD reduces the risk from 12.24 to 0.64 on average, whereas the risk under Naive KD ends at 13.27. This contrast empirically corroborates the generalization of effect alignment, and it also suggests that naive KD fails to learn from the intrinsic reasoning pattern of the teacher model.

Parametric sensitivity. Fig. 4 examines the sensitivity of RCD to the hyperparameters ρ, λ_{RCD}, and K_{eg} on the Qwen2.5-Math series. Overall, RCD is robust to the choices of ρ and λ_{RCD} because it maintains strong performance over the broad ranges $\rho \in [0.2, 1.0]$ and $\lambda_{\mathrm{RCD}} \in [0.1, 20]$. Moreover, the performance of RCD remains stable once K_{eg} exceeds 300, with only minor performance variations for $K_{\mathrm{eg}} \in [400, 1000]$. Notably, $K_{\mathrm{eg}} = 500$ attains the highest average accuracy of 75.97%. These results demonstrate that RCD does not require delicate hyperparameter tuning.

Due to space limitations, we defer to Appendix E the comparison between RCD and DistiLLM-2 (Ko et al., 2025), which uses both training data and data generated by student. Experimental results show that RCD also achieves better overall performance than DistiLLM-2 under this mixed-data setting. Additionally, we provide an analysis of performance–runtime trade-off for all methods, which shows that RCD achieves the best performance while incurring moderate computational overhead.

7 Conclusion

In this paper, we study knowledge distillation for LLMs to transfer the reasoning capability of a teacher model to a student model. We use reasoning circuits to characterize the internal computations reflecting the reasoning capability. To guide capability transfer while preserving the student's intrinsic reasoning pattern, we propose model-inherent Reasoning Circuit Distillation (RCD). RCD independently discovers faithful circuits for teacher and student via EAP-IG. It then aligns their predictive effects by matching the logit variations induced by circuit attenuation. We theoretically prove that, under mild conditions, the effect discrepancy defined over ground-truth faithful circuits vanishes as the proposed RCD loss shrinks. Experimental results also demonstrate the superiority of RCD over representative KD methods on both mathematical reasoning and code generation.

AI USE STATEMENT

In this work, we used generative AI tools to assist with identifying and reviewing related literature, preparing figures and tables, and improving the clarity and grammar of the manuscript. These tools also assisted with checking and clarifying mathematical arguments, experimental implementation, and interpretation of experimental results. The central research idea originated from the authors and was not directly proposed by generative AI. The authors retain responsibility for all scientific decisions and the final content of this work, including text, claims, and artifacts produced with AI assistance.

ETHICS STATEMENT

This work does not involve human subjects, personally identifiable information, or the collection of sensitive data. Our experiments are conducted on publicly available datasets and pretrained language models. We are not aware of any significant ethical risks arising from the methodology or experimental setting of this work.

REPRODUCIBILITY STATEMENT

We provide the methodological and implementation details necessary to reproduce RCD in the main paper and Appendix. In particular, circuit discovery and effect alignment are described in Secs. 4.1 and 4.2, with the complete algorithm in Appendix B. Proofs of the theoretical results are given in Appendix C. Moreover, detailed descriptions of datasets, training configurations, hyperparameters for all the compared methods, and evaluation protocols are provided in Appendix D. Further experimental analyses are reported in Appendices A and E.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Adithya Bhaskar, Alexander Wettig, Dan Friedman, and Danqi Chen. Finding transformer circuits with edge pruning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 18506–18534, 2024.

Nicolas Boizard, Kevin El Haddad, CELINE HUDELOT, and Pierre Colombo. Towards cross-tokenizer distillation: the universal logit distillation loss for LLMs. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Vivien Cabannes, Charles Arnal, Wassim Bouaziz, Xingyu Alice Yang, Francois Charton, and Julia Kempe. Iteration head: A mechanistic study of chain-of-thought. In *Advances in Neural Information Processing Systems*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021a.

Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5008–5017, 2021b.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

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.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235, 2023.

Cole Gibson, Wenping Cui, and Gautam Reddy. Distinct mechanisms underlying in-context learning in transformers. *arXiv preprint arXiv:2604.12151*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Michael Hanna, Ollie Liu, and Alexandre Variengien. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36:76033–76060, 2023.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. *arXiv preprint arXiv:2403.17806*, 2024.

Reilly Haskins and Benjamin Adams. Distilled circuits: a mechanistic study of internal restructuring in knowledge distillation. *arXiv preprint arXiv:2505.10822*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *Advances in Neural Information Processing Systems*, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2.5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Yeongmin Kim, Donghyeok Shin, Mina Kang, Byeonghu Na, and Il-Chul Moon. Distillation of large language models via concrete score matching. In *International Conference on Learning Representations*, volume 2026, pp. 16649–16682, 2026.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327. Association for Computational Linguistics, 2016.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *International Conference on Machine Learning*, 2025.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. In *Findings of the Association for Computational Linguistics*, pp. 25366–25394, 2025.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. In *International Conference on Learning Representations*, volume 2024, pp. 27168–27188, 2024.

Maxime Méloux, François Portet, and Maxime Peyrard. Mechanistic interpretability as statistical estimation: A variance analysis. In *International Conference on Machine Learning*, volume 306, 2026.

Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fried Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, Martin Tutek, Amir Zur, David Bau, and Yonatan Belinkov. MIB: A mechanistic interpretability benchmark. In *International Conference on Machine Learning*, volume 267, pp. 45069–45108, 2025.

Thinh On, Hengzhi Pei, Leonard Lausen, and George Karypis. Knowledge distillation for large language models through residual learning. In *International Conference on Learning Representations*, 2026.

Donghyeok Shin, Yeongmin Kim, Suhyeon Jo, Byeonghu Na, and Il chul Moon. AMiD: Knowledge distillation for LLMs with α-mixture assistant distribution. In *International Conference on Learning Representations*, 2026.

Makoto Shing, Kou Misaki, Han Bao, Sho Yokoi, and Takuya Akiba. Taid: Temporally adaptive interpolated distillation for efficient knowledge transfer in language models. In *International Conference on Learning Representations*, volume 2025, pp. 92233–92256, 2025.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 407–416, 2024.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron Wallace, and David Bau. Function vectors in large language models. In *International Conference on Learning Representations*, volume 2024, pp. 17282–17333, 2024.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. Causal mediation analysis for interpreting neural nlp: The case of gender bias. *arXiv preprint arXiv:2004.12265*, 2020.

Somin Wadhwa, Silvio Amir, and Byron C Wallace. Circuit distillation. *arXiv preprint arXiv:2509.25002*, 2025.

Guanghui Wang, Zhiyong Yang, Zitai Wang, Shi Wang, Qianqian Xu, and Qingming Huang. ABKD: Pursuing a proper allocation of the probability mass in knowledge distillation via α-β-divergence. In *International Conference on Machine Learning*, 2025a.

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. *arXiv preprint arXiv:2211.00593*, 2022.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *Advances in Neural Information Processing Systems*, 2025b.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. In *International Conference on Learning Representations*, volume 2025, pp. 62143–62156, 2025.

Zhengxuan Wu, Atticus Geiger, Joshua Rozner, Elisa Kreiss, Hanson Lu, Thomas Icard, Christopher Potts, and Noah Goodman. Causal distillation for language models. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4288–4295, 2022.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *International Conference on Learning Representations*, volume 2024, pp. 30745–30766, 2024.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. Focal and global knowledge distillation for detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4643–4652, 2022.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. MetaMath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*, 2024.

Chen Zhang, Yang Yang, Jiahao Liu, Jingang Wang, Yunsen Xian, Benyou Wang, and Dawei Song. Lifting the curse of capacity gap in distilling language models. In *Proceedings of the Association for Computational Linguistics*, pp. 4535–4553, 2023.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11953–11962, 2022.

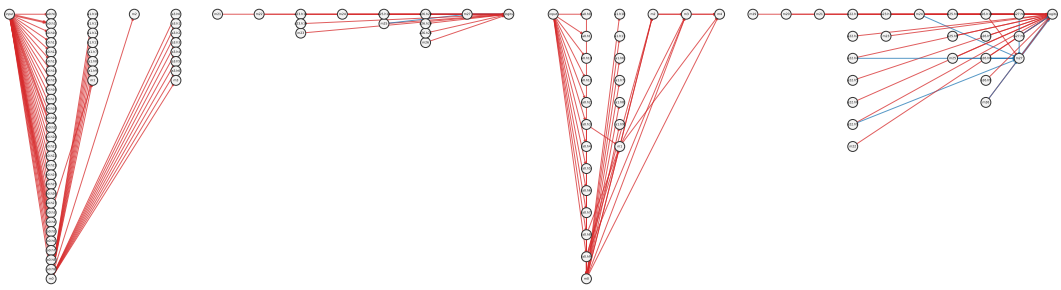
A Comparisons of Circuits for Distinct Models

In this section, we provide additional analyses and empirical results to support the claim that models with different architectures and parameter scales have distinct reasoning circuits. We first conduct qualitative and quantitative comparisons between models of different scales within the same family to examine differences in their reasoning circuits. We then extend the analysis to multiple model families, comparing circuit similarity and performing cross-task retrieval to examine whether circuits retain model-inherent structure across tasks. Together, these experiments provide empirical support for our claim that reasoning circuits differ across models.

Qualitative analysis. We first provide qualitative analysis on the circuits discovered by EAP-IG. Specifically, following the visualization protocol of Fig. 1, we retain the 80 edges with the largest absolute attribution scores in each circuit. Within each model pair, we keep the task, procedure of circuit discovery, and the number of displayed edges fixed. These visualizations reveal model-dependent differences in the locations and connectivity of edges with high attribution scores. As shown in Figs. 5–7, the discovered teacher and student circuits contain different patterns of attention and MLP connections for both mathematical reasoning and code generation. The specific differences are summarized as follows:

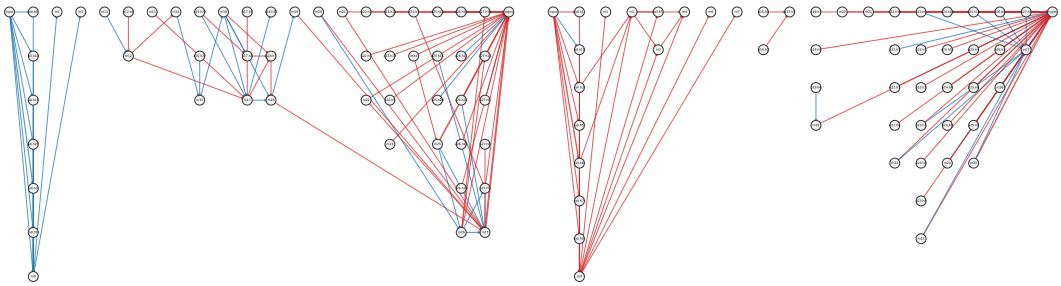
- For the Qwen2-Math series (Fig. 5), the 7B model emphasizes the direct paths from the input to the attention heads in the early layers, while the 1.5B model also uses MLP modules in the early layers and the routes from the attention heads in the late layers to the output layer.
- For the Qwen2.5-Math series (Fig. 6), although both models have 28 layers, the 7B model relies on edges that connect the intermediate MLP modules to the late MLP modules, while the 1.5B model leverages more edges from the attention heads in the late layers to the output layer than its 7B counterpart.
- For the DeepSeek-Coder series (Fig. 7), the 6.7B circuit contains prominent connections from an early MLP block to subsequent attention heads, whereas the 1.3B circuit exhibits a similar branching pattern at a later depth, with different downstream connections.

These distinctions persist between the equally deep Qwen2.5-Math models in Fig. 6, which indicates that these differences cannot be attributed solely to network depth. Together with Fig. 1, these results support our claim that reasoning circuits are model-inherent. This finding also motivates us to align the prediction effects of the discovered circuits without requiring the functional correspondence between attention heads and MLP blocks across models as in Circuit Distillation (Wadhwa et al., 2025).



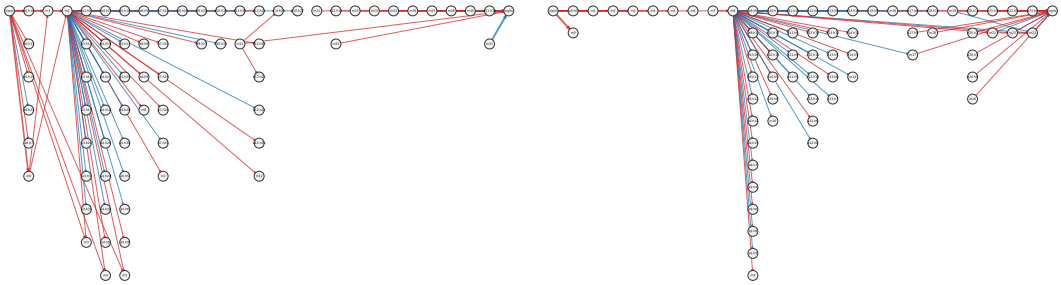
(a) Qwen2-Math-7B-Inst (teacher model) (b) Qwen2-Math-1.5B (student model)

Figure 5: The reasoning circuits of Qwen2-Math-7B-Instruct and Qwen2-Math-1.5B, each containing the 80 edges with the largest absolute values of attribution scores obtained by EAP-IG. Blue and red edges denote positive and negative attribution scores, respectively, while edge thickness represents attribution magnitude. In this figure, the 7B model concentrates more critical edges along direct paths from the input to early attention heads, whereas the 1.5B model distributes its critical edges across early attention and MLP paths as well as late attention-to-output connections.



(a) Qwen2.5-Math-7B-Inst (teacher model) (b) Qwen2.5-Math-1.5B (student model)

Figure 6: The reasoning circuits of Qwen2.5-Math-7B-Instruct and Qwen2.5-Math-1.5B. Blue and red edges denote positive and negative attribution scores, respectively, while edge thickness represents attribution magnitude. Although both models contain 28 Transformer layers, their critical edges differ substantially: the 7B model relies more on MLP chains in the intermediate and later layers, whereas the 1.5B model routes more edges from late attention heads directly to the output.



(a) DeepSeek-Coder-6.7B-Inst (teacher model) (b) DeepSeek-Coder-1.3B (student model)

Figure 7: The reasoning circuits of DeepSeek-Coder-6.7B-Instruct and DeepSeek-Coder-1.3B. Blue and red edges denote positive and negative attribution scores, respectively, while edge thickness represents attribution magnitude. In this figure, prominent MLP-to-attention connections originate near an early MLP block in the 6.7B circuit and a later block in the 1.3B circuit. The shared branching pattern and different downstream connections illustrate both common structure and model-dependent variation in the discovered circuits.

Quantitative analysis. We also conduct quantitative analysis to examine structural overlap across model scales relative to the variability of circuit discovery within each model. We quantify overlap via the Jaccard score $J(A, B) = |A \cap B|/|A \cup B|$, where A and B are the edge sets of the two circuits. For each model pair on a given task, we use the same 200 examples and five integration steps. For each model, we select the 80 edges with the largest absolute attribution scores in each mode and take their union as its task-relevant circuit. The exact-edge comparison retains all layer, module, attention-head, and Q/K/V coordinates, whereas the head-invariant comparison removes head indices and merges duplicate connections with the same source, destination, and Q/K/V type. The mappings used in both comparisons are defined in the paragraph "Mathematical definitions of circuit comparisons" below.

Tab. 3 reports average Jaccard scores of 0.198 and 0.378 for the exact-edge comparison and the head-invariant comparison, respectively. The increase in overlap after removing head indices indicates shared layer-level structure, while the overlap remains substantially lower than one. These empirical results suggest that teacher and student still differ considerably in their reasoning circuits.

Furthermore, we examine whether the observed circuit differences can be explained by sampling randomness alone. Specifically, we repeat circuit discovery on ten bootstrap samples of 200 examples for each Qwen2.5-Math model, and we select the top-300 edges by absolute attribution score in each mode to obtain the reasoning circuit. Empirically, across the $\binom{10}{2} = 45$ replicate pairs, the within-model exact-edge Jaccard scores have a mean and standard deviation of 0.834 ± 0.028 for

7B and 0.740 ± 0.039 for 1.5B. For comparison, the circuits discovered from the original samples with the same discovery procedure yield cross-model exact-edge and head-invariant scores of 0.180 and 0.394, respectively. Therefore, within-model exact-edge overlap is substantially higher than cross-model exact-edge overlap. This suggests that sampling variability alone is unlikely to account for the observed differences between the discovered teacher and student circuits.

Table 3: Cross-scale Jaccard overlap on the same task.

Model Pair	Exact	Head-Invariant
Qwen2.5-Math	0.163	0.333
Qwen2-Math	0.251	0.422
Qwen2.5-Coder	0.209	0.444
Qwen2-IF	0.168	0.311
Average	0.198	0.378

Table 4: Circuit overlap within and across models.

Comparison	Exact	Normalized
Same model	0.342	0.357
Different models	0.071	0.082

Cross-architecture analysis. We extend the comparison to eleven models from four families, spanning parameter scales from 70M to 12B, namely, Gemma-3-1B-pt, Gemma-3-4B-pt, and Gemma-3-12B-pt; Qwen3-0.6B-Base, Qwen3-1.7B, Qwen3-4B, and Qwen3-8B; Pythia-70M-deduped, Pythia-410M-deduped, and Pythia-1B-deduped; and Phi-2 (2.7B). The tasks considered are mathematical reasoning, code generation, and instruction following. Our framework distinguishes three modes introduced in Sec. 4.1, namely, prompt-to-reasoning, step-to-next-step, and reasoning-to-answer. For each model and task, we use 200 examples per mode and five integration steps for circuit discovery using the EAP-IG method. Moreover, we take the union of the per-mode top-80 edges ranked by absolute attribution scores in Eq. (3) as the discovered reasoning circuit for each model on each task.

In Tab. 4, "Same model" reports the mean overlap over 33 comparisons between different tasks within the same model. "Different models" reports the mean overlap over 495 comparisons between distinct models, including 165 same-task comparisons and 330 cross-task comparisons. The normalized comparison matches attention heads by representation similarity and groups MLP blocks by relative depth, as described in the paragraph "Representation-based normalization" below. On average, overlap between circuits from different tasks of the same model is higher than overlap between circuits from different models, under both exact-edge and normalized comparisons.

Circuits discovered on different tasks retain sufficient model-inherent structure to identify the model that produced them. To examine this, we perform cross-task retrieval: a circuit from one task is compared with the circuits of all eleven models on another task, and the model with the highest Jaccard similarity is selected. A retrieval is counted as correct only if the actual model ranks strictly above all other candidates; ties count as failures. Exact-edge and normalized retrieval achieve accuracies of 92.4% (61/66) and 97.0% (64/66), respectively, compared with the random baseline of 9.1% (1/11). Fig. 8 provides a complementary visualization of different attention and MLP connections in similarly sized Gemma-3-1B and Pythia-1B models on mathematical reasoning.

In summary, these results support the conclusion that different models have distinct reasoning circuits. This motivates us to propose model-inherent Reasoning Circuit Distillation (RCD), which aligns the predictive effects of reasoning circuits for effective KD.

Mathematical definitions of circuit comparisons. Let $a_{\ell,h}$ denote the h-th attention head in layer ℓ and let m_ℓ denote the MLP in that layer, with layers indexed from zero. The node maps for exact and head-invariant comparisons are

$$\phi^{\mathrm{ex}}(a_{\ell,h}) = (\mathrm{attn}, \ell, h), \quad \phi^{\mathrm{hi}}(a_{\ell,h}) = (\mathrm{attn}, \ell),$$
$$\phi^{\mathrm{ex}}(m_\ell) = (\mathrm{mlp}, \ell), \qquad \phi^{\mathrm{hi}}(m_\ell) = (\mathrm{mlp}, \ell).$$

Both maps preserve the distinct input and output nodes. For an edge $e = (u_e, v_e)$, let $q_e \in \{\mathrm{Q}, \mathrm{K}, \mathrm{V}, \varnothing\}$ identify the type of destination node v_e, with $\varnothing$ used for edges into MLP blocks or the output node. Given a node map ϕ, define

$$\Phi_\phi(C) = \{(\phi(u_e), \phi(v_e), q_e) : e \in C\}, \qquad J_\phi(C_A, C_B) = \frac{|\Phi_\phi(C_A) \cap \Phi_\phi(C_B)|}{|\Phi_\phi(C_A) \cup \Phi_\phi(C_B)|}.$$

The set operation merges duplicate mapped edges, and Q/K/V labels are retained throughout. Exact overlap measures agreement in architectural coordinates, while head-invariant overlap measures agreement in layer-level topology without establishing functional head correspondence.

Representation-based normalization. For each model pair (A, B), we determine head correspondence from a fixed set of $n_{\text{cal}} = 200$ mathematical reasoning examples shared by all models. Once the attention heads of two models are matched, we use this fixed matching to compare their circuits across all tasks. For $M \in \{A, B\}$ and a head $u = a_{\ell,h}$, let $R_{M,u} \in \mathbb{R}^{n_{\text{cal}} \times d_M^{\text{head}}}$ contain its output representations before the attention output projection, where d_M^{head} is the head dimension. Each row averages the head output over the text-token positions of one calibration example, excluding padding and the initial special token. Each row represents the same calibration example across models, even when their tokenizers produce different token sequences. We use linear centered kernel alignment (CKA) to compare heads with potentially different feature dimensions:

$$H = I - \frac{\mathbf{1}\mathbf{1}^\top}{n_{\text{cal}}}, \qquad G_{M,u} = HR_{M,u}R_{M,u}^\top H, \qquad S(u,v) = \frac{\langle G_{A,u}, G_{B,v}\rangle_F}{\|G_{A,u}\|_F \|G_{B,v}\|_F}.$$

Here, I is the $n_{\text{cal}} \times n_{\text{cal}}$ identity matrix, $\mathbf{1}$ is the vector of ones, and $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product. A head with a zero centered Gram matrix is assigned zero similarity.

Let U_0 and V_0 contain all attention heads of models A and B, respectively. Starting from these sets, we greedily select the most similar remaining pair and remove both heads:

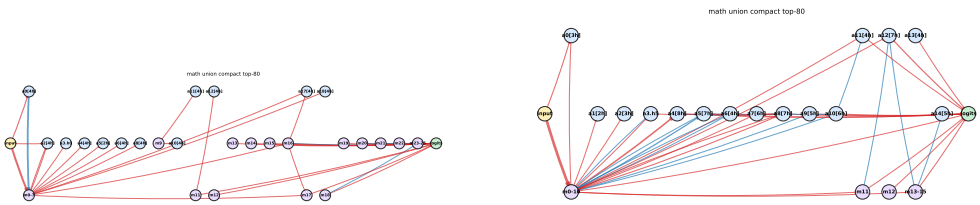
$$(u_k, v_k) \in \underset{u \in U_{k-1},\, v \in V_{k-1}}{\arg\max} S(u,v), \qquad U_k = U_{k-1} \setminus \{u_k\}, \qquad V_k = V_{k-1} \setminus \{v_k\}.$$

This procedure continues for $k = 1, \ldots, \min\{|U_0|, |V_0|\}$, with ties resolved by lexicographic layer and head indices. It yields a one-to-one matching of the smaller head set to a subset of the larger set, without reusing either endpoint. Matching considers all layers and does not assume that equal head indices have equal functions.

For this model pair, assign both heads in the k-th matched pair the common label (attn, k). An unmatched head keeps a model-specific label, so edges involving it are not counted as shared edges between the two models. MLPs are mapped by relative depth, using $b_M(\ell) = \lfloor 8\ell / L_M \rfloor$ for a model with L_M layers:

$$\phi_M^{A,B}(w) = \begin{cases} (\text{attn}, k), & w = u_k \text{ in } A \text{ or } w = v_k \text{ in } B, \\ (\text{unmatched}, M, \ell, h), & w = a_{\ell,h} \text{ is unmatched}, \\ (\text{mlp}, b_M(\ell)), & w = m_\ell, \\ w, & w \in \{\text{input}, \text{output}\}. \end{cases}$$

Normalized overlap is $J(\Phi_{\phi_A^{A,B}}(C_A), \Phi_{\phi_B^{A,B}}(C_B))$. Edges incident to unmatched heads remain in the union and therefore contribute to its denominator. When we compare two tasks of the same model, each attention head is matched to itself, while MLP blocks are still grouped by relative depth. These maps are defined separately for each model pair and are fixed independently of the circuit edges being compared. Heads are matched by representation similarity; matched heads need not have the same effects on prediction.



(a) Gemma-3-1B-pt (b) Pythia-1B-deduped

Figure 8: Task-relevant reasoning circuits (for mathematical reasoning task) from two model architectures at a similar parameter scale. Each compact circuit aggregates the union of the per-mode top-80 edges. In both figures, blue lines and red lines denote positive attribution scores and negative attribution scores, respectively, and line width represents the aggregated attribution magnitude.

B Algorithm Description for Reasoning Circuit Distillation

Algorithm 1 summarizes the complete procedure of Reasoning Circuit Distillation (RCD) introduced in Sec. 4.2.

Algorithm 1 Reasoning Circuit Distillation (RCD)

Input: The teacher model $\mathcal{M}_{\mathrm{T}}$ and the pre-trained student model $\mathcal{M}_{\mathrm{S}}$; the training dataset $\mathcal{D} = \{(x^i, y^i)\}_{i=1}^{N}$, with $X^i = (x^i, y^i)$; the dataset $\mathcal{P}$ for circuit discovery; the set of modes $\mathcal{R}$; and the base loss $\mathcal{L}_{\mathrm{base}}$.
Hyperparameters: Number of discovery pairs n; number of interpolation points m; number of selected edges K_{eg} for each mode; batch-level token budget K_H; attenuation scale ρ; Coefficient λ_{RCD} for the RCD loss function; and the number of training epochs E_{tr}.
for $M \in \{\mathrm{T}, \mathrm{S}\}$ **do**
  for each mode $r \in \mathcal{R}$ **do**
    Construct n pairs of (X, X^{cf}) from $\mathcal{P}$ according to the source and target spans of mode r;
    Compute the estimated attribution score $\widehat{s}^{\,r}_{M,e}$ for every $e \in \mathcal{E}_M$ via Eq. (3);
    Obtain the circuit $\widehat{C}^r_M \leftarrow \arg\max_{C \subseteq \mathcal{E}_M,\, |C| = K_{\mathrm{eg}}} \sum_{e \in C} |\widehat{s}^{\,r}_{M,e}|$;
  end for
  Derive the union circuit $\widehat{C}_M \leftarrow \bigcup_{r \in \mathcal{R}} \widehat{C}^r_M$;
end for
Freeze the teacher parameters and keep the edge sets $\widehat{C}_{\mathrm{T}}$ and $\widehat{C}_{\mathrm{S}}$ fixed during distillation;
for $q = 1, \ldots, E_{\mathrm{tr}}$ **do**
  for each mini-batch $\mathcal{B} \subset \mathcal{D}$ **do**
    Compute logits $\boldsymbol{z}_{M,t}(\varnothing; X^i, \rho), \forall\, X^i \in \mathcal{B}$, $M \in \{\mathrm{T}, \mathrm{S}\}$;
    Compute predictive entropy $H^i_t \leftarrow -\sum_{v \in \mathcal{V}} p^i_t(v) \log p^i_t(v)$, where $p^i_t(v)$ is the predictive probability of $\mathcal{M}_{\mathrm{T}}$;
    Choose token allocation $k_i \propto |\mathcal{T}(X^i)|$, with $\sum_{X^i \in \mathcal{B}} k_i = \min\{K_H, \sum_{X^i \in \mathcal{B}} |\mathcal{T}(X^i)|\}$;
    Selected positions $\mathcal{H}^i \leftarrow \arg\max_{\mathcal{H} \subseteq \mathcal{T}(X^i),\, |\mathcal{H}| = k_i} \sum_{t \in \mathcal{H}} H^i_t$, $\forall\, X^i \in \mathcal{B}$;
    Apply intervention $\mathrm{do}\big(e \leftarrow (1-\rho)\boldsymbol{z}^{\mathrm{out}}_{M,u_e}(X^i)\big)$ for each $e \in \widehat{C}_M$, and compute the logits $\boldsymbol{z}_{M,t}(\widehat{C}_M; X^i, \rho), \forall\, M \in \{\mathrm{T}, \mathrm{S}\}$;
    Compute circuit effects $\boldsymbol{\Delta}_{M,t}(\widehat{C}_M; X^i, \rho)$ via Eq. (5), $\forall\, t \in \mathcal{H}^i$, $M \in \{\mathrm{T}, \mathrm{S}\}$;
    Compute RCD loss $\mathcal{L}_{\mathrm{RCD}}$ over $\mathcal{B}$ via Eq. (6);
    Total loss $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{base}}(\boldsymbol{\theta}_{\mathrm{S}}) + \lambda_{\mathrm{RCD}} \mathcal{L}_{\mathrm{RCD}}$;
    Update $\boldsymbol{\theta}_{\mathrm{S}}$ by minimizing $\mathcal{L}$ with the AdamW Optimizer;
  end for
end for
Output: The distilled student model $\mathcal{M}_{\mathrm{S}}$.

An example for mode-specific circuit discovery For mathematical reasoning and code generation, our framework defines the same three modes introduced in Sec. 4.1: prompt-to-reasoning, step-to-next-step, and reasoning-to-answer. We illustrate them using a simple mathematical reasoning example. To this end, we first write the clean example as $X = (x, y) = (x, h_1, h_2, a)$, where x is the prompt, h_1 and h_2 are consecutive reasoning steps, and a is the final answer. Let $X^{\mathrm{cf}} = (x^{\mathrm{cf}}, h_1^{\mathrm{cf}}, h_2^{\mathrm{cf}}, a^{\mathrm{cf}})$ be an independently sampled example from the discovery dataset $\mathcal{P}$ that supplies counterfactual activations. The following illustrative pair makes the source and target spans concrete.

Span	Clean example X	Counterfactual example X^{cf}
Prompt	Mia has 3 packs of stickers, with 4 stickers in each pack. She gives away 5 stickers. How many stickers does she have left?	Tom has 2 bags of marbles, with 8 marbles in each bag. He gives away 3 marbles. How many marbles does he have left?
Step 1	h_1: Mia starts with $3 \times 4 = 12$ stickers.	h_1^{cf}: Tom starts with $2 \times 8 = 16$ marbles.
Step 2	h_2: After giving away 5, she has $12 - 5 = 7$ stickers.	h_2^{cf}: After giving away 3, he has $16 - 3 = 13$ marbles.
Answer	a: The answer is 7.	a^{cf}: The answer is 13.

In the displays below, blue denotes the clean source span, ochre denotes its counterfactual counterpart, and green denotes the clean target span. An arrow $s \leftarrow s^{\mathrm{cf}}$ means that the activations transmitted along the edge under evaluation at the positions of s are replaced by activations from the aligned counterfactual span s^{cf}. The text identifies the relevant positions; the intervention operates on hidden activations. Alignment is performed separately with each model's tokenizer. We choose a counterfactual span no longer than the clean source span and align their starting positions. If the counterfactual span is shorter, we append padding tokens to its end until the lengths match, assigning these padding positions an attention-mask value of zero. This preserves the positions of subsequent target tokens; counterfactual spans longer than the clean source span are excluded.

For mathematical reasoning, we separate the final answer from the reasoning trace and treat each nonempty line of the trace as a step. For code generation, we split the explanatory text into steps at blank lines, such as `\n\n`. If this yields fewer than two steps, we try numbered or bulleted list boundaries, followed by sentence boundaries. Step-to-next-step discovery uses only examples with at least two steps and selects consecutive steps in their original order.

(1) Prompt-to-reasoning: prompt information supports the reasoning trace.

$$\underbrace{x \leftarrow x^{\mathrm{cf}}}_{\text{source activation replacement}} \qquad \underbrace{(h_1, h_2)}_{\text{target tokens}} .$$

We replace the activations on edge e at Mia's prompt positions with those from Tom's prompt, and evaluate the log-likelihood of both clean reasoning steps: "Mia starts with $3 \times 4 = 12$ stickers" and "After giving away 5, she has $12 - 5 = 7$ stickers." This asks how strongly e supports the clean reasoning trace through information carried at the prompt positions.

(2) Step-to-next-step: an intermediate result supports the next step.

$$\underbrace{h_1 \leftarrow h_1^{\mathrm{cf}}}_{\text{source activation replacement}} \qquad \underbrace{h_2}_{\text{target tokens}} .$$

We replace the activations on e at "Mia starts with $3 \times 4 = 12$ stickers" with those from "Tom starts with $2 \times 8 = 16$ marbles." Only the immediately following clean step, "After giving away 5, she has $12 - 5 = 7$ stickers," contributes to the log-likelihood. This measures how replacing the activations on e at the first step's positions changes the log-likelihood of the next step. For a longer trace, the same construction uses any eligible adjacent pair (h_k, h_{k+1}), with only h_k as the source span.

(3) Reasoning-to-answer: the complete trace supports the final answer.

$$\underbrace{(h_1, h_2) \leftarrow (h_1^{\mathrm{cf}}, h_2^{\mathrm{cf}})}_{\text{source activation replacement}} \qquad \underbrace{a}_{\text{target tokens}} .$$

We replace the activations on e across the entire clean reasoning trace with those from the counterfactual trace, which concludes that Tom has 13 marbles. The target remains the clean answer, "The answer is 7." Thus, the score measures how much the clean reasoning activations on e support this answer; the counterfactual answer a^{cf} is never used as the target.

From span replacement to edge attribution. For each mode, the clean and intervened runs use the same input tokens and the same clean target labels. Only the contribution of the selected edge at the source positions is overwritten, and subsequent computations propagate this intervention normally. The exact attribution of an edge is the decrease in average target-token log-likelihood caused by patching that edge, as defined in Eq. (2). A positive attribution score means that the replacement reduces the likelihood of the clean target. EAP-IG approximates this attribution using the embedding interpolation in Eq. (3), with the clean target labels fixed along the path. Averaging over discovery pairs yields $\widehat{s}^{\mathcal{T}}_{M,e}$. We then select the K_{eg} edges with the largest $|\widehat{s}^{\mathcal{T}}_{M,e}|$ for each mode and take their union. This procedure is performed independently for $\mathcal{M}_{\mathrm{T}}$ and $\mathcal{M}_{\mathrm{S}}$.

C Proofs of Theoretical Results

In this section, we provide the proofs for Theorem 1 and Theorem 2. Specifically, in Sections C.1–C.2, we derive bounds on the attribution error terms induced by EAP-IG and then provide the circuit

recovery guarantee in Theorem 1. Additionally, in Sections C.3–C.4, we use this guarantee to bound the population discrepancy in predictive effects between the ground-truth circuits and then prove Theorem 2.

C.1 Useful Lemmas for Circuit Recovery with EAP-IG

In the following, without loss of generality, we fix a model $\mathcal{M}_M$, $M \in \{\mathrm{T}, \mathrm{S}\}$, and a mode $r \in \mathcal{R}$. We use n independent and identically distributed (IID) pairs of examples to conduct the circuit discovery process and the number of interpolation points in the EAP-IG method is defined as m. We also leave the source-position mask of each mode implicit below, as masking activation differences and gradients does not increase their norms and preserves the subsequent bounds under Assumption 1.

Throughout these proofs, the norm of an activation or gradient matrix is the Euclidean norm of its vectorization. Specifically, for $A \in \mathbb{R}^{p\times q}$, we write $\|A\|_2 := \|\operatorname{vec}(A)\|_2 = (\sum_{i=1}^{p}\sum_{j=1}^{q} A_{ij}^2)^{1/2}$, which equals its Frobenius norm. The corresponding inner product is $\langle A, B\rangle = \sum_{i,j} A_{ij}B_{ij}$.

We first explain the sources of estimation error in circuit discovery with EAP-IG. Specifically, evaluating the exact single-edge interventions in Eq. (2) requires a separate intervened forward pass for each candidate edge e, which is expensive when $E_M = |\mathcal{E}_M|$ is large. EAP-IG reduces this cost by using the shared input embedding $\mathbf{E}_M^{(\alpha)}(X, X^{\mathrm{cf}}) = (1-\alpha)\mathbf{E}_M(X^{\mathrm{cf}}) + \alpha\mathbf{E}_M(X)$, with $\alpha \in [0,1]$. Here, "shared input" means that all edge contributions are estimated with the same interpolated input at a given α. One forward and backward pass at this input supplies gradients with respect to the target-node inputs for all candidate edges. Combining these gradients with the cached activation differences $\boldsymbol{d}_{M,e}$ gives all edge contributions at a certain interpolation point. Averaging them over $\alpha = j/m$, $j = 1, \ldots, m$, gives the estimates of attribution scores for a single discovery pair. This reduces computational cost, and the resulting error between the estimated attribution score and the ground-truth attribution score is quantified in the lemmas below.

The shared embedding path differs from the exact single-edge path, which interpolates only the contribution transmitted along e, because interpolating the input embedding changes activations throughout the network. Consequently, the target-node gradient $\widetilde{\boldsymbol{g}}_{M,e}^{r}(\alpha)$ on the shared path can differ from $\boldsymbol{g}_{M,e}^{r}(\alpha)$ on the exact path. Assumption 1 controls this discrepancy through $\|\widetilde{\boldsymbol{g}}_{M,e}^{r}(\alpha) - \boldsymbol{g}_{M,e}^{r}(\alpha)\|_2 \le \kappa_M$.

For conciseness, we omit the discovery pair of examples, namely, (X, X^{cf}) in the notation. We define the exact pointwise attribution score and its counterpart with shared input as

$$\phi_{M,e}^{r}(\alpha) := \left\langle \boldsymbol{d}_{M,e}, \boldsymbol{g}_{M,e}^{r}(\alpha)\right\rangle \quad \text{and} \quad \widetilde{\phi}_{M,e}^{r}(\alpha) := \left\langle \boldsymbol{d}_{M,e}, \widetilde{\boldsymbol{g}}_{M,e}^{r}(\alpha)\right\rangle, \tag{9}$$

respectively. Here, $\boldsymbol{d}_{M,e}(X, X^{\mathrm{cf}}) \in \mathbb{R}^{|X|\times d_M}$ denotes the activation difference with the source-position mask understood as above. The fundamental theorem of calculus gives $s_{M,e}^{r} = \mathbb{E}\int_0^1 \phi_{M,e}^{r}(\alpha)d\alpha$. We also write $\widetilde{s}_{M,e}^{r} := \mathbb{E}\int_0^1 \widetilde{\phi}_{M,e}^{r}(\alpha)d\alpha$ for the corresponding population score with shared input.

The following lemmas separate three sources of estimation error, namely, (a) replacing the exact single-edge path with the shared embedding path; (b) approximating the path integral with m interpolation points; and (c) estimating the population expectation from n IID discovery pairs of examples. The first two error terms contribute bounds of $2\overline{Z}_M\kappa_M$ and $\overline{Z}_M H_M/m$, respectively, while the third is controlled by concentration. Increasing m reduces the integration error but does not, in general, eliminate the error from using the shared path.

Lemma 1 (Boundedness of activation difference). *Under Assumption 1, every candidate edge e satisfies*

$$\|\boldsymbol{d}_{M,e}(X, X^{\mathrm{cf}})\|_2 \le 2\overline{Z}_M \tag{10}$$

for every pair of discovery examples (X, X^{cf}).

Proof. Assumption 1 gives $\|\boldsymbol{z}_{M,u_e}^{\mathrm{out}}(X)\|_2 \le \overline{Z}_M$ and $\|\boldsymbol{z}_{M,u_e}^{\mathrm{out}}(X^{\mathrm{cf}})\|_2 \le \overline{Z}_M$. By the definition of $\boldsymbol{d}_{M,e}$ and the triangle inequality, we obtain

$$\|\boldsymbol{d}_{M,e}\|_2 \le \|\boldsymbol{z}_{M,u_e}^{\mathrm{out}}(X)\|_2 + \|\boldsymbol{z}_{M,u_e}^{\mathrm{out}}(X^{\mathrm{cf}})\|_2 \le 2\overline{Z}_M. \tag{11}$$

This completes the proof. □

Lemma 2 (Boundedness of Attribution integrand and shared-input approximation error). *Under Assumption 1, for every candidate edge e and all $\alpha, \alpha' \in [0,1]$, we have*

$$\begin{aligned} \max\Big\{|\phi^r_{M,e}(\alpha)|, |\widetilde{\phi}^r_{M,e}(\alpha)|\Big\} &\le 2\overline{Z}_M G_M, \\ |\widetilde{\phi}^r_{M,e}(\alpha) - \phi^r_{M,e}(\alpha)| &\le 2\overline{Z}_M \kappa_M, \\ |\widetilde{\phi}^r_{M,e}(\alpha) - \widetilde{\phi}^r_{M,e}(\alpha')| &\le 2\overline{Z}_M H_M |\alpha - \alpha'|. \end{aligned} \tag{12}$$

Consequently, it also holds that $|\widetilde{s}^r_{M,e} - s^r_{M,e}| \le 2\overline{Z}_M \kappa_M$.

Proof. We use the following three conditions from Assumption 1, which is valid for every edge e and all $\alpha, \alpha' \in [0,1]$:

$$\begin{aligned} \max\big\{\|\boldsymbol{g}^r_{M,e}(\alpha)\|_2, \|\widetilde{\boldsymbol{g}}^r_{M,e}(\alpha)\|_2\big\} &\le G_M, \\ \|\widetilde{\boldsymbol{g}}^r_{M,e}(\alpha) - \boldsymbol{g}^r_{M,e}(\alpha)\|_2 &\le \kappa_M, \\ \|\widetilde{\boldsymbol{g}}^r_{M,e}(\alpha) - \widetilde{\boldsymbol{g}}^r_{M,e}(\alpha')\|_2 &\le H_M |\alpha - \alpha'|. \end{aligned} \tag{13}$$

Combining each condition with $\|\boldsymbol{d}_{M,e}\|_2 \le 2\overline{Z}_M$ from Lemma 1, we apply the Cauchy–Schwarz inequality to obtain the three corresponding bounds in Eq. (12). Finally, we integrate the approximation error over α and take the expectation to obtain

$$|\widetilde{s}^r_{M,e} - s^r_{M,e}| \le \mathbb{E} \int_0^1 |\widetilde{\phi}^r_{M,e}(\alpha) - \phi^r_{M,e}(\alpha)| d\alpha \le 2\overline{Z}_M \kappa_M. \tag{14}$$

This completes the proof. □

Lemma 3 (Discretization error between the EAP-IG estimate and the shared-input integral). *Under Assumption 1, for an integer $m \ge 1$, we define the EAP-IG estimate with m interpolation points for each pair of examples (X, X^{cf}) as $Z^{(r,m)}_{M,e}(X, X^{\mathrm{cf}})$, given by $Z^{(r,m)}_{M,e}(X, X^{\mathrm{cf}}) := \frac{1}{m}\sum_{j=1}^m \widetilde{\phi}^r_{M,e}\big(\frac{j}{m}\big)$. Then we have $|Z^{(r,m)}_{M,e}| \le 2\overline{Z}_M G_M$ and*

$$\left| Z^{(r,m)}_{M,e} - \int_0^1 \widetilde{\phi}^r_{M,e}(\alpha) d\alpha \right| \le \frac{\overline{Z}_M H_M}{m}. \tag{15}$$

Proof. We first obtain the magnitude bound using Eq. (12), which gives $|\widetilde{\phi}^r_{M,e}(\alpha)| \le 2\overline{Z}_M G_M$. Next, to bound the discretization error, we set $I_j = [(j-1)/m, j/m]$. The condition $\|\widetilde{\boldsymbol{g}}^r_{M,e}(\alpha) - \widetilde{\boldsymbol{g}}^r_{M,e}(\alpha')\|_2 \le H_M|\alpha - \alpha'|$ in Assumption 1 yields $|\widetilde{\phi}^r_{M,e}(\alpha) - \widetilde{\phi}^r_{M,e}(\alpha')| \le 2\overline{Z}_M H_M |\alpha - \alpha'|$. Applying this bound on each interval I_j, we obtain

$$\left| Z^{(r,m)}_{M,e} - \int_0^1 \widetilde{\phi}^r_{M,e}(\alpha) d\alpha \right| \le \sum_{j=1}^m \int_{I_j} 2\overline{Z}_M H_M \left(\frac{j}{m} - \alpha\right) d\alpha = \frac{\overline{Z}_M H_M}{m}. \tag{16}$$

This completes the proof. □

Lemma 4 (Uniform concentration of EAP-IG estimates across edges). *Under Assumption 1, for n IID pairs of discovery examples, if we have $\overline{Z}_M G_M > 0$, then for every $t > 0$, it holds that*

$$\Pr\left(\max_{e \in \mathcal{E}_M} \left| \frac{1}{n} \sum_{i=1}^n Z^{(r,m)}_{M,e}(X_i, X^{\mathrm{cf}}_i) - \mathbb{E} Z^{(r,m)}_{M,e} \right| \ge t \right) \le 2E_M \exp\left(-\frac{nt^2}{8\overline{Z}^2_M G^2_M} \right). \tag{17}$$

When $\overline{Z}_M G_M = 0$, the sampling deviation is exactly zero.

Proof. Lemma 3 shows that $Z_{M,e}^{(r,m)} \in [-2\overline{Z}_M G_M, 2\overline{Z}_M G_M]$. Therefore, for a certain edge e, the n estimates for each pair of discovery examples, namely, (X, X^{cf}) are IID because the discovery pairs are IID. Therefore, for $\overline{Z}_M G_M > 0$ and $t > 0$, Hoeffding's inequality yields

$$\begin{aligned}\Pr\left(\left|\frac{1}{n}\sum_{i=1}^{n} Z_{M,e}^{(r,m)}(X_i, X_i^{\mathrm{cf}}) - \mathbb{E}Z_{M,e}^{(r,m)}\right| \geq t\right) \\ \leq 2\exp\left(-\frac{2n^2t^2}{n(4\overline{Z}_M G_M)^2}\right) = 2\exp\left(-\frac{nt^2}{8\overline{Z}_M^2 G_M^2}\right).\end{aligned} \tag{18}$$

Next, we take a union bound over the E_M candidate edges to obtain Eq. (17). If $\overline{Z}_M G_M = 0$, Lemma 3 instead gives $Z_{M,e}^{(r,m)} = 0$ for every edge, proving the remaining case. □

Lemma 5 (Uniform error bound on attribution scores of EAP-IG). *For any $\delta \in (0,1)$, we define*

$$\varepsilon_M(n, m, \delta) := 2\overline{Z}_M \kappa_M + \frac{\overline{Z}_M H_M}{m} + 2\overline{Z}_M G_M \sqrt{\frac{2\log(2E_M/\delta)}{n}}. \tag{19}$$

Under Assumption 1, for n IID discovery pairs, it holds that

$$\Pr\left(\max_{e \in \mathcal{E}_M} |\widehat{s}_{M,e}^r - s_{M,e}^r| \leq \varepsilon_M(n, m, \delta)\right) \geq 1 - \delta. \tag{20}$$

Proof. By Eq. (3), we have $\widehat{s}_{M,e}^r = n^{-1}\sum_{i=1}^n Z_{M,e}^{(r,m)}(X_i, X_i^{\mathrm{cf}})$. For each edge, we add and subtract $\mathbb{E}Z_{M,e}^{(r,m)}$ and $\widetilde{s}_{M,e}^r$ to obtain

$$|\widehat{s}_{M,e}^r - s_{M,e}^r| \leq \left|\frac{1}{n}\sum_{i=1}^{n} Z_{M,e}^{(r,m)}(X_i, X_i^{\mathrm{cf}}) - \mathbb{E}Z_{M,e}^{(r,m)}\right| + \left|\mathbb{E}Z_{M,e}^{(r,m)} - \widetilde{s}_{M,e}^r\right| + |\widetilde{s}_{M,e}^r - s_{M,e}^r|. \tag{21}$$

We next use Lemmas 2 and 3 to bound the sum of the last two terms by $\overline{Z}_M H_M/m + 2\overline{Z}_M \kappa_M$. For the sampling term, if $\overline{Z}_M G_M = 0$, Lemma 4 ensures that it is zero. Otherwise, we set $t = 2\overline{Z}_M G_M \sqrt{2\log(2E_M/\delta)/n}$ in Lemma 4 to obtain a uniform bound with probability at least $1-\delta$. Finally, we substitute these bounds into Eq. (21) to establish the claim in both cases. This completes the proof. □

C.2 Proof of Theorem 1

Proof. We fix $M \in \{\mathrm{T}, \mathrm{S}\}$ and $\delta \in (0,1)$ and consider n IID discovery pairs. For each mode r, C_M^r denotes the set of K_{eg} edges with the largest absolute population scores $|s_{M,e}^r|$, whereas $\widehat{C}_M^r$ uses the estimated scores. Therefore, $|C_M^r| = |\widehat{C}_M^r| = K_{\mathrm{eg}}$, with $1 \leq K_{\mathrm{eg}} < E_M$. Under Assumptions 1 and 2, we have

$$\gamma_{M,r} := \min_{e \in C_M^r} |s_{M,e}^r| - \max_{e \notin C_M^r} |s_{M,e}^r| > 0 \quad \text{and} \quad C_M^\star = \bigcup_{r \in \mathcal{R}} C_M^r. \tag{22}$$

Theorem 1 further requires the adjusted margin $\underline{\gamma}_M := \min_{r \in \mathcal{R}}(\gamma_{M,r} - 4\overline{Z}_M \kappa_M) > 0$. Its conditions on the number of interpolation points m and the number of discovery pairs n are

$$m > \frac{4\overline{Z}_M H_M}{\underline{\gamma}_M}, \quad \text{and} \quad n > \frac{128\overline{Z}_M^2 G_M^2 \log(2E_M|\mathcal{R}|/\delta)}{\underline{\gamma}_M^2}, \tag{23}$$

respectively. We first set $\eta := \delta/|\mathcal{R}| \in (0,1)$ to control the failure probability for each transition mode r. Since $\log(2E_M/\eta) = \log(2E_M|\mathcal{R}|/\delta)$, the two conditions above imply

$$\frac{2\overline{Z}_M H_M}{m} < \frac{\underline{\gamma}_M}{2}, \quad \text{and} \quad 4\overline{Z}_M G_M \sqrt{\frac{2\log(2E_M/\eta)}{n}} < \frac{\underline{\gamma}_M}{2}, \tag{24}$$

respectively. The second term is zero when $\overline{Z}_M G_M = 0$. Subsequently, Eq. (19) and Eq. (24) give

$$2\varepsilon_M(n, m, \eta) < 4\overline{Z}_M \kappa_M + \underline{\gamma}_M \leq \gamma_{M,r}. \tag{25}$$

Next, we show that this error bound preserves the top-K_{eg} selection. By Lemma 5, the event $\max_{e\in\mathcal{E}_M}|\widehat{s}^r_{M,e}-s^r_{M,e}|\le\varepsilon_M(n,m,\eta)$ has probability at least $1-\eta$. On this event, the reverse triangle inequality gives $\big||\widehat{s}^r_{M,e}|-|s^r_{M,e}|\big|\le\varepsilon_M(n,m,\eta)$ for every edge. As a result, for every $e\in C^r_M$ and $f\notin C^r_M$, we obtain

$$\begin{aligned}|\widehat{s}^r_{M,e}|-|\widehat{s}^r_{M,f}|&\ge|s^r_{M,e}|-|s^r_{M,f}|-2\varepsilon_M(n,m,\eta)\\&\ge\gamma_{M,r}-2\varepsilon_M(n,m,\eta)>0.\end{aligned}\tag{26}$$

Both C^r_M and $\widehat{C}^r_M$ contain the K_{eg} edges with the largest absolute population attribution score $s^r_{M,e}$ and the estimated attribution score $\widehat{s}^r_{M,e}$, respectively. Consequently, no edge crosses the top-K_{eg} boundary, and $\widehat{C}^r_M=C^r_M$ with probability at least $1-\eta$. Finally, we apply this result to every mode and take a union bound to obtain simultaneous mode recovery with probability at least $1-|\mathcal{R}|\eta=1-\delta$. Based on this condition and Assumption 2, it directly holds that

$$\widehat{C}_M=\bigcup_{r\in\mathcal{R}}\widehat{C}^r_M=\bigcup_{r\in\mathcal{R}}C^r_M=C^\star_M.\tag{27}$$

This completes the proof. □

C.3 Useful Lemmas for Effect Alignment between Ground-Truth Reasoning Circuits

In this section, we use $C^\star_M$ and $\widehat{C}_M$ to denote the ground-truth circuit and the discovered circuit, respectively, and the sets of their edges are kept fixed when evaluating circuit effects. Predictive effects are evaluated at fixed teacher and student checkpoints and a fixed attenuation scale $\rho\in[0,1]$. Additionally, the $N\ge1$ alignment examples are IID and independent of training, model selection, and circuit discovery. We omit $\boldsymbol{\theta}_{\mathrm{S}}$ inside circuit effects indexed by S when it is clear from context.

For the generalization analysis, high-entropy response tokens are selected separately for each example. Specifically, $\mathcal{H}(X)$ contains the $k(X)$ response positions with the highest predictive entropy under the fixed teacher, where $1\le k(X)\le|\mathcal{T}(X)|$ is a deterministic function of X alone and ties are resolved by a fixed rule. Consequently, $\mathcal{H}(X)$ does not depend on other alignment examples. This per-example selection convention is used in the proof, and the training algorithm allocates its token budget within each mini-batch. Here, $\mathcal{L}_{\mathrm{RCD}}$ denotes the empirical average of Eq. (6) over all N alignment examples, with $\mathcal{H}^i=\mathcal{H}(X^i)$.

Lemma 6 (**Failure probability for model-inherent circuit recovery**). *Under Assumptions 1 and 2, let* $a_{M,r}=\gamma_{M,r}/2-2\overline{Z}_M\kappa_M-\overline{Z}_MH_M/m>0$ *and define*

$$p_{M,r}:=2E_M\exp\left(-\frac{na^2_{M,r}}{8\overline{Z}^2_MG^2_M}\right).\tag{28}$$

Then $\Pr(\widehat{C}^r_M\ne C^r_M)\le p_{M,r}$. *Moreover, defining* $p_M:=\sum_{r\in\mathcal{R}}p_{M,r}$, *we have*

$$\Pr(\widehat{C}_M\ne C^\star_M)\le p_M.\tag{29}$$

Proof. We first bound the failure probability for an individual mode. If the empirical top-K_{eg} set differs from its population counterpart, the ordering argument in Eq. (26) implies $\max_e|\widehat{s}^r_{M,e}-s^r_{M,e}|\ge\gamma_{M,r}/2$. The condition $a_{M,r}=\gamma_{M,r}/2-2\overline{Z}_M\kappa_M-\overline{Z}_MH_M/m>0$ ensures that the deterministic error bound $2\overline{Z}_M\kappa_M+\overline{Z}_MH_M/m$ is strictly smaller than $\gamma_{M,r}/2$. It also implies $\gamma_{M,r}>0$. Therefore, Lemma 2 implies $\overline{Z}_MG_M>0$, since otherwise every population score and the margin would be zero. By the decomposition in Eq. (21) and the two deterministic bounds in Lemmas 2 and 3, we derive that the event of a recovery failure is contained in the following event:

$$\left\{\max_e\left|\frac1n\sum_{i=1}^n Z^{(r,m)}_{M,e}(X_i,X^{\mathrm{cf}}_i)-\mathbb{E}Z^{(r,m)}_{M,e}\right|\ge a_{M,r}\right\}.\tag{30}$$

We therefore substitute $t=a_{M,r}$ into Eq. (17) to establish the first claim. Next, we consider recovery of the union of the mode-specific circuits. If every mode-specific circuit is recovered, Assumption 2 gives $\widehat{C}_M=C^\star_M$. Therefore, $\{\widehat{C}_M\ne C^\star_M\}\subseteq\bigcup_{r\in\mathcal{R}}\{\widehat{C}^r_M\ne C^r_M\}$, and we apply a union bound to obtain Eq. (29). This completes the proof. □

Lemma 7 (Discrepancy of predictive effects between circuits of the same model). *For a fixed attenuation scale $\rho \in [0,1]$, let $\boldsymbol{z}_{M,t}(C;X,\rho)$ denote the output-logit vector obtained by jointly attenuating all edges in C at scale ρ, so $\boldsymbol{\Delta}_{M,t}(C;X,\rho) = \boldsymbol{z}_{M,t}(\varnothing;X,\rho) - \boldsymbol{z}_{M,t}(C;X,\rho)$. Under Assumption 1, for any circuits C, C', input X, and selected position t,*

$$\|\boldsymbol{\Delta}_{M,t}(C;X,\rho) - \boldsymbol{\Delta}_{M,t}(C';X,\rho)\|_2 \le 2\overline{Z}_M. \tag{31}$$

Proof. The clean terms cancel, so $\boldsymbol{\Delta}_{M,t}(C;X,\rho) - \boldsymbol{\Delta}_{M,t}(C';X,\rho) = \boldsymbol{z}_{M,t}(C';X,\rho) - \boldsymbol{z}_{M,t}(C;X,\rho)$. Assumption 1 gives $\|\boldsymbol{z}_{M,t}(C;X,\rho)\|_2 \le \overline{Z}_M$ and $\|\boldsymbol{z}_{M,t}(C';X,\rho)\|_2 \le \overline{Z}_M$. The triangle inequality then gives Eq. (31). This completes the proof. □

Lemma 8 (Upper bound on the population effect alignment risk of discovered circuits). *For the discovered circuits $\widehat{C}_{\mathrm{T}}$ and $\widehat{C}_{\mathrm{S}}$, define the population effect alignment risk as*

$$R_{\mathrm{disc}}(\boldsymbol{\theta}_{\mathrm{S}};\rho) := \mathbb{E}_X\left[\frac{1}{|\mathcal{H}(X)|}\sum_{t\in\mathcal{H}(X)} \|\boldsymbol{\Delta}_{\mathrm{T},t}(\widehat{C}_{\mathrm{T}};X,\rho) - \boldsymbol{\Delta}_{\mathrm{S},t}(\widehat{C}_{\mathrm{S}};X,\rho)\|_2\right]. \tag{32}$$

Under Assumption 1, we fix the teacher model, the student model, and the attenuation scale $\rho \in [0,1]$. Let $\mathcal{L}_{\mathrm{RCD}}(\boldsymbol{\theta}_{\mathrm{S}};\rho)$ be the RCD loss over N IID alignment examples, which are independent of the data used for training, model selection, and circuit discovery. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta/2$, it holds that

$$R_{\mathrm{disc}}(\boldsymbol{\theta}_{\mathrm{S}};\rho) \le \mathcal{L}_{\mathrm{RCD}}(\boldsymbol{\theta}_{\mathrm{S}};\rho) + 2(\overline{Z}_{\mathrm{T}} + \overline{Z}_{\mathrm{S}})\sqrt{\frac{\log(2/\delta)}{2N}}. \tag{33}$$

Proof. Assumption 1 ensures $\|\boldsymbol{z}_{M,t}(C;X,\rho)\|_2 \le \overline{Z}_M$ for every $C \subseteq \mathcal{E}_M$, including $C = \varnothing$. By the triangle inequality, we have $\|\boldsymbol{\Delta}_{M,t}(C;X,\rho)\|_2 \le 2\overline{Z}_M$. Therefore, each per-example average in Eq. (32) lies in $[0,B]$, where $B = 2(\overline{Z}_{\mathrm{T}} + \overline{Z}_{\mathrm{S}})$. Next, we condition on the discovery samples and the outcomes of training and model selection, so the circuits and checkpoints are fixed. Token selection depends only on each example, making every per-example average the same deterministic function of its own input. Since the alignment examples remain IID under this conditioning, these averages are IID with expectation $R_{\mathrm{disc}}(\boldsymbol{\theta}_{\mathrm{S}};\rho)$ and empirical mean $\mathcal{L}_{\mathrm{RCD}}(\boldsymbol{\theta}_{\mathrm{S}};\rho)$. The stop-gradient operation does not alter the numerical value of this empirical mean. We then apply the one-sided Hoeffding inequality with failure probability $\delta/2$ to obtain Eq. (33). Because the conditional bound holds for every realization of the discovery samples and the training and model-selection outcomes, it also holds unconditionally. This completes the proof. □

Lemma 9 (Effect discrepancy between the discovered circuit and the ground-truth circuit). *For each $\mathcal{M} \in \{\mathcal{M}_{\mathrm{T}}, \mathcal{M}_{\mathrm{S}}\}$, define the random population discrepancy*

$$D_M(\rho) := \mathbb{E}_X\left[\frac{1}{|\mathcal{H}(X)|}\sum_{t\in\mathcal{H}(X)} \|\boldsymbol{\Delta}_{M,t}(C_M^\star;X,\rho) - \boldsymbol{\Delta}_{M,t}(\widehat{C}_M;X,\rho)\|_2\right]. \tag{34}$$

Under Assumptions 1 and 2, suppose that $a_{M,r} = \gamma_{M,r}/2 - 2\overline{Z}_M\kappa_M - \overline{Z}_M H_M/m > 0$ for both $M \in \{\mathrm{T},\mathrm{S}\}$ and every $r \in \mathcal{R}$. With n IID discovery pairs for each model, Lemma 6 then gives

$$\mathbb{E}[D_M(\rho)] \le 0 \cdot \Pr(\widehat{C}_M = C_M^\star) + 2\overline{Z}_M \Pr(\widehat{C}_M \ne C_M^\star) \le 2\overline{Z}_M p_M. \tag{35}$$

Consequently, with probability at least $1-\delta/2$,

$$D_{\mathrm{T}}(\rho) + D_{\mathrm{S}}(\rho) \le \frac{4}{\delta}\left(\overline{Z}_{\mathrm{T}} p_{\mathrm{T}} + \overline{Z}_{\mathrm{S}} p_{\mathrm{S}}\right). \tag{36}$$

Proof. On the event $\{\widehat{C}_M = C_M^\star\}$, we have $D_M(\rho) = 0$. Otherwise, Lemma 7 bounds every summand in Eq. (34) by $2\overline{Z}_M$, and hence $D_M(\rho) \le 2\overline{Z}_M$. Taking expectations over the discovery samples, we obtain

$$\mathbb{E}[D_M(\rho)] \le 2\overline{Z}_M \Pr(\widehat{C}_M \ne C_M^\star) \le 2\overline{Z}_M p_M, \tag{37}$$

where the last step uses Eq. (29). Finally, we sum the expectations for $\mathcal{M}_{\mathrm{T}}$ and $\mathcal{M}_{\mathrm{S}}$ and apply Markov's inequality with failure probability $\delta/2$ to obtain Eq. (36). This completes the proof. □

C.4 Proof of Theorem 2

Proof. We fix $\delta \in (0,1)$ and $\rho \in [0,1]$ and use the reference circuits and evaluation checkpoints specified in Section C.3. Each model uses n IID discovery pairs, and the N IID alignment examples are independent of training, model selection, and circuit discovery. Under Assumptions 1 and 2, Theorem 2 further requires

$$a_{M,r} := \frac{\gamma_{M,r}}{2} - 2\overline{Z}_M \kappa_M - \frac{\overline{Z}_M H_M}{m} > 0, \qquad \forall\, M \in \{\mathrm{T},\mathrm{S}\},\ r \in \mathcal{R}. \tag{38}$$

Therefore, the deviation threshold for sampling is positive for every model and every mode, and then the bound on recovery failure is explicitly

$$p_M := \sum_{r\in\mathcal{R}} 2E_M \exp\left(-\frac{n a_{M,r}^2}{8\overline{Z}_M^2 G_M^2}\right) = \sum_{r\in\mathcal{R}} p_{M,r}, \qquad \forall\, M \in \{\mathrm{T},\mathrm{S}\}. \tag{39}$$

These conditions allow us to apply Lemmas 6–9 to both models. Specifically, we first decompose the discrepancy between the effects of ground-truth circuits using the discovered circuit effects. For every realization of the discovery samples, we apply the triangle inequality to obtain

$$R_{\mathrm{GT}}(\boldsymbol{\theta}_{\mathrm{S}};\rho) \le R_{\mathrm{disc}}(\boldsymbol{\theta}_{\mathrm{S}};\rho) + D_{\mathrm{T}}(\rho) + D_{\mathrm{S}}(\rho). \tag{40}$$

Specifically, we insert the two discovered-circuit effects between the ground-truth effects of $\mathcal{M}_{\mathrm{T}}$ and $\mathcal{M}_{\mathrm{S}}$, apply the inequality pointwise, and then average over selected positions and X.

Next, we use Lemma 8 to bound the first term on the right-hand side of Eq. (40) with failure probability $\delta/2$. We use Lemma 9 to bound the sum of the remaining two terms with the same failure probability. By a union bound, both inequalities hold with probability at least $1-\delta$. Note that this step does not require the two events to be independent. Finally, we substitute these bounds into Eq. (40) to obtain

$$R_{\mathrm{GT}}(\boldsymbol{\theta}_{\mathrm{S}};\rho) \le \mathcal{L}_{\mathrm{RCD}}(\boldsymbol{\theta}_{\mathrm{S}};\rho) + 2(\overline{Z}_{\mathrm{T}} + \overline{Z}_{\mathrm{S}})\sqrt{\frac{\log(2/\delta)}{2N}} + \frac{4}{\delta}\left(\overline{Z}_{\mathrm{T}} p_{\mathrm{T}} + \overline{Z}_{\mathrm{S}} p_{\mathrm{S}}\right), \tag{41}$$

which is Eq. (8). This completes the proof. □

D Detailed Experimental Setup

In this section, we provide further details on the datasets, training procedures, and evaluation protocols used in our experiments.

D.1 Dataset Description

We provide detailed descriptions of the datasets used for distillation and evaluation below.

- **MetaMathQA (mathematical reasoning).** MetaMathQA (Yu et al., 2024) contains approximately 395K mathematical query–response pairs. It is constructed by bootstrapping existing mathematical problems from multiple perspectives, including forward reasoning, backward reasoning, and question rephrasing. We randomly select 50K valid pairs to form the fixed distillation set.
- **GSM8K (mathematical reasoning).** GSM8K (Cobbe et al., 2021) comprises approximately 8.5K linguistically diverse grade-school word problems, each of which typically requires multiple reasoning steps involving elementary arithmetic. We evaluate all distilled models on its complete test split of 1,319 problems.
- **MATH (mathematical reasoning).** MATH (Hendrycks et al., 2021) contains 12.5K competition-level mathematical problems covering seven subjects and five difficulty levels. Each problem is accompanied by a step-by-step solution. We use the complete test split of 5,000 problems for evaluation.
- **WizardCoder (code generation).** The WizardCoder corpus (Luo et al., 2024) is constructed by applying Evol-Instruct (Xu et al., 2024) to Code Alpaca, progressively increasing instruction complexity by adding constraints, deepening reasoning requirements, and imposing efficiency requirements. The resulting corpus contains approximately 78K evolved code instruction–response pairs. After removing invalid examples, we randomly select approximately 50K pairs as the fixed distillation set.

- **HumanEval and HumanEval+ (code generation).** HumanEval (Chen et al., 2021a) contains 164 hand-written Python programming problems. Each problem specifies a function through its signature and docstring and is paired with hidden unit tests for functional-correctness evaluation. HumanEval+ retains the same problems while substantially extending their test suites. We evaluate on all 164 problems and report both HumanEval and HumanEval+ results following the EvalPlus protocol (Liu et al., 2023).

- **MBPP and MBPP+ (code generation).** Mostly Basic Python Problems (MBPP) (Austin et al., 2021) contains approximately 1,000 crowd-sourced Python problems designed for entry-level programmers. Each problem provides a natural-language task description, a reference solution, and three test cases. MBPP+ supplements the original problems with more comprehensive tests; we report both MBPP and MBPP+ results under the EvalPlus protocol (Liu et al., 2023).

D.2 Details for Training and Evaluation

Training details. Unless otherwise specified, we follow the experimental setup of DistiLLM-2 (Ko et al., 2025). In detail, we truncate each training sequence, including the prompt and response, to at most 1,024 tokens and compute the training loss only on response tokens. We use Low-Rank Adaptation (LoRA) (Hu et al., 2022) with rank 16, scaling parameter 128, and dropout 0.05 for the query, key, value, and output projections in each attention layer and the gate, up, and down projections in each MLP layer. In each experiment, the student model is trained for $E_{\mathrm{tr}} = 2$ epochs in bfloat16 precision using the AdamW optimizer with a learning rate of 5×10^{-5} and zero weight decay. We employ a cosine learning-rate schedule with a warmup ratio of 0.1. The per-device batch size and number of gradient accumulation steps are adjusted according to the memory requirement of each method while keeping the effective batch size at 128. Distillation training is performed on four NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs. The SFT baseline method uses the full task-specific corpora rather than the 50K distillation subsets. Specifically, the mathematical reasoning models are fine-tuned on the full MetaMathQA corpus for one epoch with a maximum sequence length of 2,048 and a learning rate of 2×10^{-5}. Additionally, the SFT models for code generation are fine-tuned on the full WizardCoder corpus for two epochs with a maximum sequence length of 2,048 and a learning rate of 5×10^{-6}.

For all the baseline methods, we follow the recommended settings in the corresponding papers and released implementations. The hyperparameters adopted for these methods are summarized in Tab. 5. These settings are shared across the four model pairs. Moreover, the parameters that are not listed in the table follow the common training configuration described above. Following standard KD practice (Hinton et al., 2015; Kim & Rush, 2016), Naive KD and Naive KD + RCD use $\mathcal{L}_{\mathrm{base}} = \mathcal{L}_{\mathrm{CE}} + \mathcal{L}_{\mathrm{KD}}$, where $\mathcal{L}_{\mathrm{CE}}$ is the cross-entropy loss and $\mathcal{L}_{\mathrm{KD}}$ is defined in Eq. (1).

For the two variants of our method (namely, Naive KD + RCD and ABKD + RCD), we compute the base KD loss on the response tokens with a maximum sequence length of 1,024. We use a batch-level token budget of $K_H = 96$ and select the response positions with the highest predictive entropy under $\mathcal{M}_{\mathrm{T}}$. At each selected position, we compute and align circuit effects only over the 10 vocabulary items assigned the highest probabilities by $\mathcal{M}_{\mathrm{T}}$, as low-probability tokens are unlikely to be generated next, and thus including them in the RCD loss may introduce irrelevant training signals. The RCD variants built on ABKD use $(\alpha, \beta) = (0.2, 0.7)$, which is also adopted by the ABKD method in Tab. 5. For circuit discovery, we use $n = 1{,}000$ example pairs per mode and $m = 5$ interpolation points. Additionally, for each teacher–student pair and each RCD variant, we evaluate the final checkpoint of each candidate configuration on a task-specific held-out validation set of 200 examples drawn from MetaMathQA for mathematical reasoning or WizardCoder for code generation. Each validation set is disjoint from the corresponding distillation training set. We select the configuration with the highest pass@1 accuracy for mathematical reasoning. Moreover, for code generation, we select the configuration with the lowest validation cross-entropy loss on reference response tokens. Tab. 6 reports the hyperparameter configurations selected by the validation procedure, where K_{eg}, ρ, and λ_{RCD} follow the definitions in Sec. 4.1 and Sec. 4.2.

Table 5: Hyperparameters adopted in our experiments for the baseline methods. A dash indicates that no additional hyperparameters are introduced.

Method	**Hyperparameters**
SFT	–
Naive KD (Hinton et al., 2015)	KD temperature $\tau = 1$; CE/KD loss weights $\lambda_{\mathrm{CE}} = \lambda_{\mathrm{KD}} = 1$
SeqKD (Kim & Rush, 2016)	–
ABKD (Wang et al., 2025a)	parameters for α-β-divergence $(\alpha, \beta) = (0.2, 0.7)$
ResidualKD (On et al., 2026)	residual-loss coefficient $\lambda = 0.5$; alignment-space dimension $d_A = 64$; projector pre-training $= 10$ epochs; projector learning rate $= 10^{-3}$; projector weight decay $= 10^{-4}$
Causal Distillation (Wu et al., 2022)	KD temperature $\tau = 1$; CE/KD/causal loss weights $\lambda_{\mathrm{CE}} = \lambda_{\mathrm{KD}} = \lambda_{\mathrm{causal}} = 1$; alignment scope = all layers; ratio of intervention tokens $= 0.3$
Circuit Distillation (Wadhwa et al., 2025)	KD temperature $\tau = 1$; CE/KD/CKA loss weights $\lambda_{\mathrm{CE}} = \lambda_{\mathrm{KD}} = \lambda_{\mathrm{CKA}} = 1$; number of functional-mapping examples $= 200$; mapping batch size $= 16$
CSM (Kim et al., 2026)	CSD weighting functions $(w_1, w_2) = (p_{\mathrm{T}}, p_{\mathrm{S}})$; KD temperature $\tau = 1$; CE/KD/CSD loss weights $\lambda_{\mathrm{CE}} = \lambda_{\mathrm{KD}} = \lambda_{\mathrm{CSD}} = 1$
AMiD (Shin et al., 2026)	mixture-geometry parameter $\alpha = -3$ for Qwen2-Math and -5 otherwise; mixing coefficient $\lambda = 0.1$; parameters for α-β-divergence $(\alpha_{\mathrm{AB}}, \beta_{\mathrm{AB}}) = (0.2, 0.7)$
ULD (Boizard et al., 2025)	model temperatures $\tau_{\mathrm{T}} = \tau_{\mathrm{S}} = 1$; ULD distillation factor $\lambda = 1.5$

Table 6: Hyperparameters adopted for the proposed RCD variants, *i.e.*, Naive KD + RCD and ABKD + RCD.

Model pair	$\mathcal{L}_{\mathbf{base}}$	$K_{\mathbf{eg}}$	ρ	$\lambda_{\mathbf{RCD}}$
Qwen2.5-Math-7B-Instruct → Qwen2.5-Math-1.5B	Naive KD	500	0.2	15
	ABKD	500	0.2	0.2
Qwen2-Math-7B-Instruct → Qwen2-Math-1.5B	Naive KD	500	1.0	0.3
	ABKD	500	1.0	0.05
Qwen2.5-Coder-7B-Instruct → Qwen2.5-Coder-1.5B	Naive KD	200	0.8	0.02
	ABKD	500	1.0	0.015
DeepSeek-Coder-6.7B-Instruct → DeepSeek-Coder-1.3B	Naive KD	200	1.0	0.02
	ABKD	800	0.8	0.02

Evaluation details. We first merge each LoRA adapter into its initial checkpoint and perform generation with vLLM in bfloat16 precision on a single GPU. For mathematical reasoning, we use the official Qwen chain-of-thought chat template[1], whose system instruction asks the model to reason step by step and place its final answer within `\boxed{}`. We use greedy decoding with temperature 0 and top-$p = 1$, generating one response per problem with at most 512 new tokens for GSM8K and 1,024 for MATH. For GSM8K, we extract the final numerical answer from the boxed expression or final-answer marker and compare it with the reference answer after removing formatting differences such as commas. For MATH, we extract the final boxed expression and determine correctness after LaTeX normalization, numerical comparison, and symbolic-equivalence checking. For code generation, we use the direct-completion prompts provided by EvalPlus (Liu et al., 2023) and greedily generate one completion per problem with at most 1,024 new tokens. EvalPlus executes the generated programs against the base and augmented test suites for all 164 HumanEval and 378 sanitized MBPP problems. We report pass@1 accuracy on HumanEval, HumanEval+, MBPP, and MBPP+, where a completion is counted as correct only if it passes all the corresponding tests. For the main

[1] `https://github.com/QwenLM/Qwen2.5-Math`

Table 7: Comparison with DistiLLM-2 on mathematical reasoning and code generation. We report the pass@1 accuracy (%), and within each SGO block, the better result in each column is shown in **bold**.

Method	Qwen2-Math-7B-Instruct → Qwen2-Math-1.5B		
	GSM8K	MATH	Avg.
$\mathcal{M}_{\mathrm{T}}$	87.95	70.70	79.32
$\mathcal{M}_{\mathrm{S}}$	31.69	20.48	26.09
Fixed responses and SGOs			
DistiLLM-2 (Ko et al., 2025)	**79.15**	55.56	67.36
DistiLLM-2 + RCD	79.00	**56.48**	**67.74**

Method	DeepSeek-Coder-6.7B-Instruct → DeepSeek-Coder-1.3B				
	HumanEval	HumanEval+	MBPP	MBPP+	Avg.
$\mathcal{M}_{\mathrm{T}}$	71.34	62.20	77.78	67.46	69.69
$\mathcal{M}_{\mathrm{S}}$	32.93	26.83	57.41	48.15	41.33
Fixed responses and SGOs					
DistiLLM-2 (Ko et al., 2025)	37.80	32.32	**60.85**	48.94	44.98
DistiLLM-2 + RCD	**38.41**	**32.93**	60.32	**49.47**	**45.28**
Fixed responses only					
ABKD + RCD	38.41	31.71	62.96	52.12	46.30

experiments, we report the mean and standard deviation over three random seeds for each distillation method. For each seed, we compute the arithmetic mean of the accuracies across the evaluation benchmarks for each model pair. The Avg. column reports the mean and standard deviation of these per-seed averages across the three seeds.

E Additional Ablation Studies

Comparison with methods using Student-Generated Outputs (SGOs). To assess whether RCD also improves distillation with student-generated outputs, we integrate it into DistiLLM-2 (Ko et al., 2025), which is commonly adopted as a baseline method for distillation with student-generated outputs. In our experiments, the tasks considered are mathematical reasoning and code generation. For each query, we compute the RCD losses on the fixed training response and the student-generated output, denoted by $\mathcal{L}_{\mathrm{RCD}}^{\mathrm{fix}}$ and $\mathcal{L}_{\mathrm{RCD}}^{\mathrm{SGO}}$, respectively. Let w_{fix} and w_{SGO} denote the weights assigned to the corresponding distillation losses in DistiLLM-2. We use the same weights to construct the combined objective $\mathcal{L} = \mathcal{L}_{\mathrm{D2}} + \lambda_{\mathrm{RCD}}(w_{\mathrm{fix}}\mathcal{L}_{\mathrm{RCD}}^{\mathrm{fix}} + w_{\mathrm{SGO}}\mathcal{L}_{\mathrm{RCD}}^{\mathrm{SGO}})$, where $\mathcal{L}_{\mathrm{D2}}$ is the original objective of DistiLLM-2. Empirically, we set $(K_{\mathrm{eg}}, \rho, \lambda_{\mathrm{RCD}}) = (200, 0.2, 0.2)$ for the Qwen2-Math model series and $(K_{\mathrm{eg}}, \rho, \lambda_{\mathrm{RCD}}) = (200, 0.5, 0.03)$ for the DeepSeek-Coder model series.

The experimental results are provided in Tab. 7, which shows that DistiLLM-2 + RCD achieves the highest average accuracy in both settings among the compared methods that use SGOs. Specifically, RCD improves the average accuracy of DistiLLM-2 from 67.36% to 67.74% for mathematical reasoning and from 44.98% to 45.28% for code generation. These results demonstrate that RCD also benefits distillation with student-generated outputs. For code generation, ABKD + RCD trained only on fixed data achieves a higher average accuracy of 46.30%. It outperforms both DistiLLM-2 and DistiLLM-2 + RCD, which use fixed responses together with SGOs.

Runtime analysis. The proposed RCD requires one additional forward pass through each model to measure the logit variations induced by attenuating the reasoning circuit. Despite this additional computation, the computational overhead of our RCD remains competitive when compared with other baseline methods. Fig. 9 compares the performance and runtime for all methods, both averaged across the four teacher–student model pairs. In this figure, we can observe that ABKD + RCD achieves the highest overall performance of 62.50 with an average runtime of 1.49 hours, substantially faster than Causal Distillation (1.88 hours), ResidualKD (2.45 hours), and SFT (2.76 hours). It therefore lies on the empirical Pareto frontier and offers a favorable trade-off between performance and runtime.

Notably, the reported runtime for our method excludes the time spent on circuit discovery. For a fixed task and discovery configuration, the circuit characterizes the pretrained teacher's capabilities and can be reused across distillation runs because the parameters of the teacher model remain fixed. For the Qwen2.5-Math model series, circuit discovery across all three modes took about 17 min for the 1.5B model and 1 h 14 min for the 7B model.

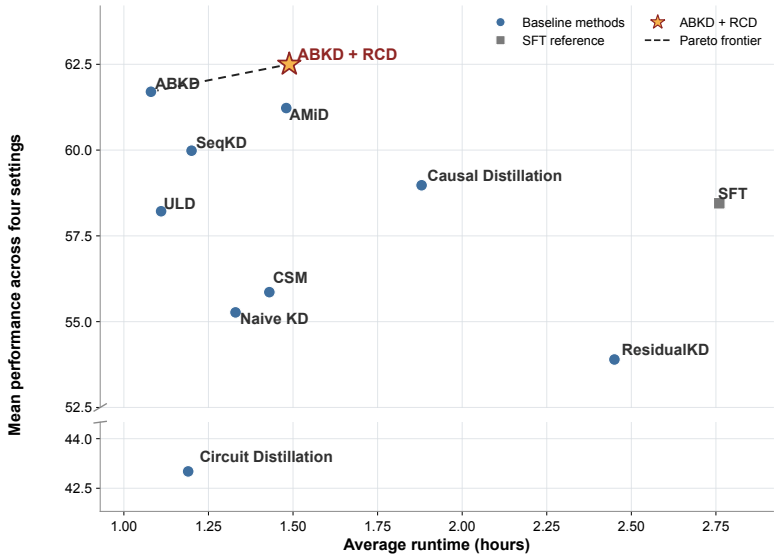


Figure 9: Performance-runtime trade-off averaged across the four model configurations. The star denotes ABKD + RCD, and the dashed line denotes the empirical Pareto frontier.