REFORMING THE MECHANISM: EDITING REASONING PATTERNS IN LLMS WITH CIRCUIT RESHAPING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) often exhibit flawed reasoning ability that undermines reliability. Existing approaches to improving reasoning typically treat it as a general and monolithic skill, applying broad training which is inefficient and unable to target specific reasoning errors. We introduce ***Reasoning Editing***, a paradigm for selectively modifying specific reasoning patterns in LLMs while preserving other reasoning pathways. This task presents a fundamental trade-off between *Generality*, the ability of an edit to generalize across different tasks sharing the same reasoning pattern, and *Locality*, the ability to preserve other reasoning capabilities. Through systematic investigation, we uncover the ***Circuit-Interference Law***: Edit interference between reasoning patterns is proportional to the overlap of their neural circuits. Guided by this principle, we propose ***REdit***, the first framework to actively reshape neural circuits before editing, thereby modulating interference between reasoning patterns and mitigating the trade-off. REdit integrates three components: (i) *Contrastive Circuit Reshaping*, which directly addresses the generality-locality trade-off by disentangling overlapping circuits; (ii) *Meta-Contrastive Learning*, which extends transferability to novel reasoning patterns; and (iii) *Dual-Level Protection*, which preserves preexisting abilities by constraining reshaping update directions and regularizing task-level predictions. Extensive experiments with `Qwen-2.5-3B` on propositional logic reasoning tasks across three difficulty levels demonstrate that REdit consistently achieves superior generality and locality compared to baselines, with additional validation in mathematics showing broader potential. Our code is available at https://anonymous.4open.science/r/REdit-DBD8.

1 INTRODUCTION

Large language models (LLMs) achieve state-of-the-art performance across various domains such as mathematics (Liu et al., 2023a; 2024a), law (Cheong et al., 2024; Sun, 2023), and medicine (Zhao et al., 2023; Hadi et al., 2023). The success arises from their exceptional reasoning ability when executing complex instructions (Lu et al., 2023; Villalobos et al., 2022; Wang et al., 2024b). Despite this success, LLMs often produce incorrect or misleading responses (Perković et al., 2024; Huang et al., 2025) driven by spurious reasoning processes, which significantly undermines their reliability and safety. For example, an LLM may correctly encode the fact that "`if a brain aneurysm is present, a CT scan will show bleeding or swelling (A→B OR A→C)`", but still wrongly infer "`no bleeding implies no aneurysm (¬B → ¬A)`", risking harmful medical consequences (Sim & Chen, 2024). Addressing such gaps remains a critical challenge for researchers and practitioners alike.

To strengthen reasoning, researchers typically view it as one general, monolithic skill that calls for broad enhancement (Wang et al., 2023; Parmar et al., 2024; Wan et al., 2024). Standard approaches include fine-tuning on large reasoning corpora (Zhang et al., 2024a; Kumar et al., 2025), reinforcement learning from human feedback (RLHF) (Havrilla et al., 2024a; Yue et al., 2025), and sophisticated test-time prompting (Bi et al., 2024; Zhang et al., 2022). However, treating the LLM's reasoning as a monolithic ability has several drawbacks. First, overall reasoning enhancement can be difficult and expensive, demanding extensive human annotation and huge computational budgets (Luo et al., 2024; Lai et al., 2025). Second, growing evidence indicates that LLMs' reasoning is not monolithic but can be decomposed into separable patterns (Zhang et al., 2025a; Jiang et al., 2025; Zhang et al., 2025c; Shao & Cheng, 2025). Indiscriminately training over every reasoning pattern fails to distinguish between those the model already handles well and those it struggles with, thus leading to inefficient use of resources and suboptimal correction of specific reasoning errors. Therefore, recent approaches have shifted towards enhancement at the level of specific reasoning trajectories or intermediate steps, which involve only a handful of reasoning patterns (Cui et al., 2025; Havrilla et al., 2024b). However, these methods heavily rely on the model's own self-verification often without the model truly mastering the correct reasoning patterns, thus failing to reliably remedy reasoning errors. As a result, how to correct erroneous and inject new reasoning patterns without retraining on the whole reasoning datasets still remains an open problem. Recent work has demonstrated that specific reasoning patterns are encoded in localized parameters or neural circuits within LLMs (Hong et al., 2024; Kim et al., 2024), mirroring the way factual knowledge is stored in model weights (Meng et al., 2022a; Yao et al., 2024; Zhang et al., 2024c). Given the success of parameter-based methods for editing piecewise knowledge in LLMs (De Cao et al., 2021; Meng et al., 2022b), we propose a natural extension: ***If knowledge can be edited through parameter modification, can we analogously edit LLMs to correct flawed reasoning patterns or inject new ones?***

In this paper, we take an initial step toward reasoning editing, defined as the selective modification of a certain LLM's reasoning pattern while preserving its factual knowledge and other reasoning pathways. To establish a rigorous foundation for this investigation, we focus on propositional logic (PL), where reasoning patterns can be precisely defined and systematically evaluated. Although structurally simple, reasoning editing in PL remains challenging due to two fundamental desiderata (Hua et al., 2024; Sun, 2025): ***(1) Generality***, edits applied to one instance should consistently generalize to all instances with the same reasoning pattern across domains, rather than memorizing surface semantics. For example, editing the transitive rule "`A→B,B→C⇒A→C`" in math should also hold in medicine. ***(2) Locality***, edits must remain narrowly scoped, correcting the targeted inference rule without impairing the LLMs' performance on other reasoning patterns it already handles correctly. For example, editing the spurious rule "`¬B→¬A⇒A→B`" should not affect modus tollens "`(A→B,¬B)⇒¬A`".The two desiderata constitute a trade-off as shown in Section 2.2, whereby enhancing one dimension typically diminishes the other, thus presenting a significant dilemma.

To tackle this trade-off, we first probe the mechanism underlying reasoning edits. Motivated by evidence that reasoning mechanism can be faithfully revealed by neural circuits, we conduct a systematic investigation into the relationship of edit effects and the circuit of reasoning pattern. Through this analysis, we discover a fundamental principle we term the ***Circuit-Interference Law***: the degree to which an edit to one reasoning pattern affects another is directly proportional to the overlap between their respective neural circuits. Guided by this observation, we introduce ***REdit***, the first framework to actively reshape circuits prior to reasoning editing, enabling controlled modulation of interference among reasoning patterns. REdit employs three key components: At its core, (1) *Contrastive Circuit Reshaping* directly addresses the generality–locality trade-off by disentangling overlapping circuits to reduce cross-reasoning pattern interference which improves locality while consolidating pattern-specific circuits to promote within-reasoning pattern generality. Building upon this foundation, (2) *Meta-Contrastive Learning* enhances transfer to broader reasoning patterns beyond those observed during reshaping and (3) *Dual-Level Protection* safeguards preexisting reasoning abilities by constraining reshaping update directions via soft null-space projection and regularizing prediction distributions of reasoning tasks. After reshaping, widely used LoRA-based editing (Ge et al., 2024) suffices to achieve the desired generality and locality. We conduct extensive experiments on `Qwen-2.5-3B` across three propositional-logic difficulty levels, showing that REdit consistently enhances *generality* while reinforcing *locality*, surpassing strong baselines. Furthermore, additional evaluations in the mathematics domain highlight REdit's potential to generalize effectively to broader reasoning scenarios. ***Our contributions can be summarized as follows:***

- **Reasoning Editing Paradigm:** We introduce the first systematic framework for reasoning editing, extending model editing from knowledge correction to the selective modification of logical inference patterns, and formally identify the generality-locality trade-off.
- **Circuit Reshaping Methodology:** We pioneer active neural circuit modulation in LLMs, enabling principled and targeted modification of specific reasoning pathways through controlled modulation rather than passive circuit analysis.
- **Novel REdit Framework:** We propose a unified approach that synergistically combines contrastive circuit shaping, meta-contrastive learning, and dual-level protection to simultaneously achieve both broad generality and precise locality in reasoning editing.

- **Empirical Validation:** We demonstrate consistent improvements on propositional logic reasoning tasks across three difficulty levels, showing superior performance in generality and locality compared to existing editing methods.

2 Preliminaries

2.1 Problem Formulation

We study the problem of reasoning editing for LLMs in the context of propositional logic. Our goal is to enable precise modifications to an LLM's reasoning behavior, ensuring it adheres to desired logical rules while preserving its existing correct ones. To formalize this, we first introduce the necessary components of propositional logic reasoning, then define reasoning patterns and their neural approximations, and finally present the reasoning editing problem.

Notations. Let $\mathcal{X} = \{x_1, \ldots, x_m\}$ be a finite set of propositional variables (PVs), each taking a truth value in $\{\text{TRUE}, \text{FALSE}\}$. Let $\mathcal{S}$ denote a fixed set of logical connectives (e.g., $\neg, \wedge, \vee, \rightarrow$). A *premise set* $\mathcal{P}$ is a collection of well-formed formulas over $(\mathcal{X}, \mathcal{S})$ that we assume to be true. A *goal* $\mathcal{G}$ is a formula over $(\mathcal{X}, \mathcal{S})$. We use the standard entailment relation $\models$ where $\mathcal{P} \models \varphi$ means every model that satisfies $\mathcal{P}$ also satisfies φ. We write $\mathcal{Y} = \{\text{TRUE}, \text{FALSE}, \text{N/A}\}$ for the three-way status labels for $\mathcal{G}$, where N/A means "neither entailed nor refuted."

Definition 1 (Propositional-Logic (PL) Reasoning) *Given premises $\mathcal{P}$ and a goal $\mathcal{G}$, infer the status of $\mathcal{G}$ as (1) "*TRUE*" if $\mathcal{P} \models \mathcal{G}$, (2) "*FALSE*" if $\mathcal{P} \models \neg\mathcal{G}$, and (3) "*N/A*" otherwise.*

Definition 2 (Reasoning Pattern) *Let $\widehat{\mathcal{X}}$ be a finite set of placeholder PVs composed of symbols with no semantic meaning. A reasoning pattern is $\pi = (\mathcal{P}(\widehat{\mathcal{X}}, S), \mathcal{G}(\widehat{\mathcal{X}}, S))$, where $\mathcal{P}(\widehat{\mathcal{X}}, S)$ is a set of premises comprising placeholders and the connectives in S and $\mathcal{G}(\widehat{\mathcal{X}}, S)$ is the goal.*

A *substitution* $\sigma : \widehat{\mathcal{X}} \rightarrow \mathcal{X}$ replaces each placeholder by a concrete PV, yielding the instantiated pair

$$\pi_\sigma \;=\; (\mathcal{P}_\sigma, \mathcal{G}_\sigma) \;=\; (\mathcal{P}(\sigma(\widehat{\mathcal{X}}), S), \mathcal{G}(\sigma(\widehat{\mathcal{X}}), S)),$$

where $\sigma(\widehat{\mathcal{X}})$ denotes the set of ground variables obtained by applying σ to each placeholder. Two instances π_σ and $\pi_{\sigma'}$ are said to *share the same reasoning pattern* exactly when they both derive from the same template π under different substitutions $\sigma \neq \sigma'$. In practice, LLMs internalize reasoning rather than executing explicit symbolic logic, formalized as neural approximation.

Definition 3 (Neural Approximation of PL) *A parameterized language model f_θ approximates PL reasoning by mapping a concrete pair $(\mathcal{P}_\sigma, \mathcal{G}_\sigma)$ to a predicted status, as $f_\theta : (\mathcal{P}_\sigma, \mathcal{G}_\sigma) \mapsto \; \widehat{y} \in \mathcal{Y}$.*

Problem 1 (Reasoning Editing) *Suppose we have a fixed neural reasoner f_θ with parameters θ, we also possess a finite* revision dataset $\mathcal{D} \;=\; \{\, (\mathcal{P}^{(i)}, \, \mathcal{G}^{(i)}, \, \widehat{y}^{(i)}, \, y^{*(i)})\}_{i=1}^{N}$ *in which each $(\mathcal{P}^{(i)}, \mathcal{G}^{(i)})$ is a concrete premise–goal pair, $\widehat{y}^{(i)} \;=\; f_\theta(\mathcal{P}^{(i)}, \mathcal{G}^{(i)})$ is the original model's prediction on that pair, and $y^{*(i)}$ is the* target status *we wish the edited model $f_{\theta'}$ to produce instead. Our objective of reasoning editing is to find a revised parameter vector θ' that meets below three requirements.*

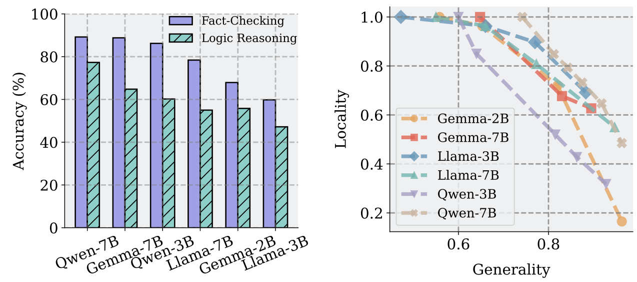
(1) Edit Success. *For each sample $(\mathcal{P}^{(i)}, \mathcal{G}^{(i)}, \widehat{y}^{(i)}, y^{*(i)})$ in $\mathcal{D}$, the edited model with parameter θ' must predict exactly the desired status, shown as $f_{\theta'}(\mathcal{P}^{(i)}, \, \mathcal{G}^{(i)}) = y^{*(i)}$.*

(2) Generality. *Let $\pi^{(i)}$ denote the underlying reasoning pattern of the example $(\mathcal{P}^{(i)}, \mathcal{G}^{(i)})$. Once we decide to revise f's behavior on one specific instantiation of $\pi^{(i)}$, we require that the edit extend to all other premise–goal pairs arising from the same abstract pattern. Formally, for any substitution σ that produces the pair $\pi_\sigma^{(i)} = (\mathcal{P}_\sigma, \mathcal{G}_\sigma)$, the edited model must satisfy $f_{\theta'}(\pi_\sigma^{(i)}) = y^{*(i)}$.*

(3) Locality. *Finally, let $\mathcal{C} \;=\; \{\, (\mathcal{P}, \mathcal{G}) \mid f_\theta(\mathcal{P}, \mathcal{G}) = y^*\}$ be the collection of all premise–goal pairs on which the original model's prediction $f_\theta(\mathcal{P}, \mathcal{G})$ already matches the ground truth. We demand that editing θ into θ' does not disturb any of these previously correct predictions. Equivalently, for every $(\mathcal{P}, \mathcal{G}) \in \mathcal{C}$, $f_{\theta'}(\mathcal{P}, \, \mathcal{G}) = f_\theta(\mathcal{P}, \, \mathcal{G})$.*

2.2 Preliminary Study

We begin by conducting preliminary experiments on a subset of propositional logic dataset ContextHub (Hua et al., 2024), where we empirically reveal a *generality–locality* trade-off: a simple edit cannot simultaneously maximize both desiderata. Our investigation is motivated by a key observation that LLMs generally lack logical reasoning ability. As Figure 1a shows, the accuracy of LLMs answering propositional logical questions (*Reasoning*) is on average 10% lower than tasks that merely require recalling the premise from the propositional logic (*Fact-Checking*). This gap highlights a systematic weakness in basic logical inference and motivates direct edits to correct faulty reasoning patterns.



(a) Fact-Checking vs. Reasoning (b) Generality-locality trade-off

Figure 1: LLM reasoning deficiencies and editing trade-off.

To evaluate whether a simple edit can achieve the dual desiderata of *generality* and *locality*, we conduct experiments to measure the two metrics. Let Π denote the index set of reasoning patterns. For each $i \in \Pi$, let $\mathcal{S}_i$ denote its instance set. Given an instance $s \in \mathcal{S}_i$, fine-tune the model on the triple $\mathcal{D}_{i,s} = (\mathcal{P}^{(s)}, \mathcal{G}^{(s)}, y^{*(s)})$ to obtain edited parameters $\theta^{(i,s)}$. The two metrics are defined as:

$$\text{Generality} = \frac{1}{\sum_i |\mathcal{S}_i|} \sum_i \sum_{s \in \mathcal{S}_i} \frac{1}{|\mathcal{S}_i \backslash \{s\}|} \sum_{(\mathcal{P},\mathcal{G}) \in \mathcal{S}_i \backslash \{s\}} \mathbb{1}[f_{\theta^{(i,s)}}(\mathcal{P}, \mathcal{G}) = y^*(\mathcal{P}, \mathcal{G})]. \tag{1}$$

$$\text{Locality} = \frac{1}{\sum_i |\mathcal{S}_i|} \sum_i \sum_{s \in \mathcal{S}_i} \frac{1}{|\Pi \setminus \{i\}|} \sum_{j \neq i} \frac{1}{|\mathcal{S}_j|} \sum_{(\mathcal{P},\mathcal{G}) \in \mathcal{S}_j} \mathbb{1}[f_{\theta^{(i,s)}}(\mathcal{P}, \mathcal{G}) = y^*(\mathcal{P}, \mathcal{G})]. \tag{2}$$

In practice, we approximate the last summation by randomly sampling a small subset of instances from each $\mathcal{S}_j$ instead of evaluating over the entire set for efficiency. We conduct experiments on multiple training configurations with learning rates $\eta \in [1\times10^{-5}, 2\times10^{-4}]$. As shown in Figure 1b, increasing η improves generality but decreases locality, yielding a trade-off between generality and locality. The remainder of this work therefore proposes an framework designed to mitigate the observed trade-off, thus leading to better editing generality while preserving locality.

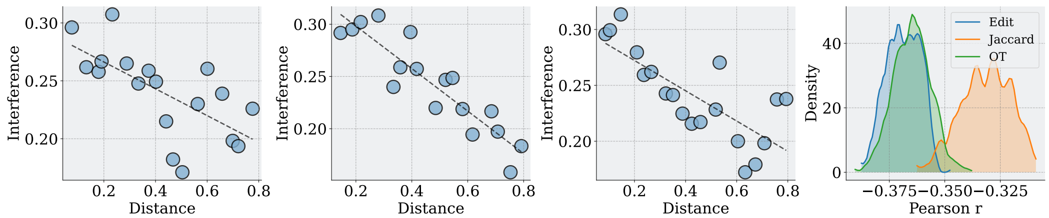
3 Methodology

3.1 Circuit-Interference Law

Prior sections reveal a generality-locality trade-off: edits often fail to generalize within the intended reasoning pattern or inadvertently spill over to other ones. To understand this gap, we turn to investigate the underlying mechanisms of reasoning editing of LLMs. Recent work in mechanistic interpretability suggests that reasoning patterns are implemented by different neural circuits, and that different tasks may recruit shared modular circuits (He et al.). Building on these findings, we conjecture that the degree of overlap or separation among these circuits may govern whether edits can generalize and remain local. Intuitively, if two reasoning patterns share substantial circuit components, editing one should also influence the other; if their circuits are largely disjoint, edits are expected to remain localized. This motivates our *central hypothesis*: circuit similarity predicts cross-pattern editing effects, with closer circuits yielding stronger interference and more distant circuits preserving locality. To validate this hypothesis, we design a four-step experimental procedure.

(1) Circuit Attribution via Edge Attribution Patching (EAP) (Syed et al., 2023). For each pattern π, we sample K instantiations $\{(\mathcal{P}_{\sigma_k}, \mathcal{G}_{\sigma_k})\}_{k=1}^{K}$ as clean input d_k^{clean} and build corrupted input d_k^{patch} detailed in Appendix E. Let $s_\theta(d)$ denote the log-probability of the ground-truth label $y^*(d)$. For an edge in the computational graph e with activation v_e, its *edge attribution* for instance k is an approximation of the score drop when e alone is patched:

$$\text{EAP}_k(e) = \langle \nabla_{v_e} s_\theta(d_k^{\text{clean}}),\ v_e(d_k^{\text{patch}}) - v_e(d_k^{\text{clean}}) \rangle.$$



(a) Edit (b) Jaccard (c) OT (d) Pearson Correlation

Figure 2: Correlation between circuit distance and interference. (a–c) Scatter plots with regression lines show that larger distances consistently correspond to reduced interference across different distance metrics. (d) Density plots of Pearson correlations confirm consistent negative associations.

To mitigate instance-specific noise unrelated to the reasoning pattern, we average the edge attributions across K instantiations, yielding $w_\pi(e) = -\frac{1}{K}\sum_{k=1}^{K} \mathrm{EAP}_k(e)$. We then define the threshold $t_\pi(\tau) = \mathrm{Quantile}_{1-\tau}(\{w_\pi(e)\})$, and construct the attributed circuit as the top–τ edges: $\mathcal{C}_\pi^{(\tau)} = \{(e, w_\pi(e)) : w_\pi(e) \geq t_\pi(\tau)\}$.

(2) Circuit Distance. Given two patterns π_i, π_j with attributed circuits $\mathcal{C}_i^{(\tau)}, \mathcal{C}_j^{(\tau)}$, we quantify structural dissimilarity using three complementary metrics: weighted edit distance $d_{Edit}(i,j)$, Jaccard distance $d_{Jaccard}(i,j)$, and optimal transport distance $d_{OT}(i,j)$ detailed in Appendix B.

(3) Interference from Single-Pattern Edits. Pick a source pattern i and a small revision set $\mathcal{D}_i = \{(\mathcal{P}^{(n)}, \mathcal{G}^{(n)}, y^{*(n)})\}_{n=1}^{N_i}$, where each $(\mathcal{P}^{(n)}, \mathcal{G}^{(n)})$ is an instance of π_i and $y^{*(n)}$ its ground truth. Obtain edited parameters $\theta_{\text{edit}(i)}$ by fine-tuning f_θ on $\mathcal{D}_i$. For any target pattern j, define accuracy on its held-out set $\mathcal{S}_j$ as $\mathrm{Acc}_j(\theta)$ and corresponding *edit interference* from i to j as $\Delta_{i\to j}$.

$$\mathrm{Acc}_j(\theta) = \frac{1}{|\mathcal{S}_j|} \sum_{(\mathcal{P},\mathcal{G})\in\mathcal{S}_j} \mathbb{1}[f_\theta(\mathcal{P},\mathcal{G}) = y^*(\mathcal{P},\mathcal{G})], \quad \Delta_{i\to j} = |\,\mathrm{Acc}_j(\theta_{\text{edit}(i)}) - \mathrm{Acc}_j(\theta)|.$$

(4) Circuit–Interference Relation. We examine the correlation between interference $\Delta_{i\to j}$ and circuit distance $d(i,j) \in \{d_{\text{Jac}}, d_{\text{Edit}}, d_{\text{OT}}\}$, modeled as $\Delta_{i\to j} \approx \alpha + \beta\, d(i,j) + \epsilon$ (Figure 2a–c). We consistently find $\beta < 0$ and negative Pearson correlations, robust across edit budgets, random seeds, and dataset subsamples as illustrated in Figure 2d. We term this finding as **Circuit–Interference Law**, which posits a monotone relationship between structural proximity and cross-pattern effects where smaller circuit distance implies larger Δ, and vice versa.

3.2 REdit: Circuit Reshaping for Reasoning Editing

The **Circuit–Interference Law** suggests that achieving both generality and locality requires well-structured circuits: representations of the same reasoning pattern should align closely, while those of different patterns should remain distinct. This leads us to a bold proposition: rather than passively analyzing existing circuits, can we actively reshape them to enforce these properties? In this paper, we take a step in that direction with **REdit**, a framework that reformulates model circuits through a contrastive meta-learning objective with dual-level protection constraints before reasoning editing, enabling more effective and controlled reasoning edits.

Contrastive Circuit Reshaping. Directly reshaping two circuits to make them similar is challenging since (i) circuit structure is discrete and (ii) circuits are not available in closed form. We therefore adopt the *attribution weights* defined in Section 3.1 as a differentiable surrogate. Within each minibatch, we sample multiple instantiations per pattern and compute their weights w_π. We then normalize them as $\tilde{w}_\pi = w_\pi / \|w_\pi\|_2$. For each anchor example i, we construct a positive example i^+ from a different group of instantiations of the same pattern, and negatives $\mathcal{N}(i)$ from instantiations of other patterns. We then conduct InfoNCE (Oord et al., 2018) over attribution vectors:

$$\mathcal{L}_{\text{ctr}}(\theta) = -\sum_i \log \frac{\exp(\langle \tilde{w}_i, \tilde{w}_{i^+}\rangle / \tau_t)}{\exp(\langle \tilde{w}_i, \tilde{w}_{i^+}\rangle / \tau_t) + \sum_{j\in\mathcal{N}(i)} \exp(\langle \tilde{w}_i, \tilde{w}_j\rangle / \tau_t)} \tag{3}$$

where τ_t is temperature. Optimizing equation 3 increases similarity within a reasoning pattern and decreases similarity across patterns, shaping circuits implicitly through their attributions.

Meta-Contrastive Learning. Training only on observed reasoning patterns may hinder transfer to rare or unseen ones. To address this, we adopt a first-order meta-learning scheme on the contrastive objective inspired by the Meta-Contrastive Network (Lin et al., 2021), adopting a Reptile-like framework (Nichol & Schulman, 2018) that iteratively samples mini-batches, performs several inner gradient steps, and updates parameters toward task-adapted weights. By aligning gradients across tasks, this process amplifies updates along shared directions while suppressing instance-specific directions, thereby mitigating overfitting to spurious contrastive relationships between particular reasoning patterns and enabling circuits to generalize beyond those observed during training. In practice, at each meta-iteration, we sample a batch of contrastive tuples $\mathcal{B}$ each regarded as a task, perform s inner steps of adaptation, and obtain task-specific parameters $\phi_i = \theta_i^s$. The outer update then moves the model weights toward the mean of these task-adapted parameters:

$$\textbf{Inner: } \theta_i^{t+1} = \theta_i^t - \alpha\,\nabla_\theta \mathcal{L}_{\text{ctr}}^{(i)}(\theta_i^t), \quad \theta_i^0 = \theta, \quad \textbf{Outer: } \theta \;\leftarrow\; \theta + \eta \cdot \tfrac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}(\phi_i - \theta). \tag{4}$$

Dual-Level Protection. To preserve the model's original behavior while enforcing correct mechanisms, we impose constraints at both the *(a) prediction level* and the *(b) optimization level*.

(a) Prediction Distribution Preservation. Given a correctness set $\mathcal{C}$ and a frozen reference model $f_{\theta^{\text{ref}}}$ (a pre-iteration snapshot of θ), we penalize deviations on $\mathcal{C}$:

$$\mathcal{L}_{\text{pred}}(\theta) = \mathbb{E}_{(\mathcal{P},\mathcal{G})\in\mathcal{C}} \mathrm{KL}\big(f_{\theta^{\text{ref}}}(\cdot \mid \mathcal{P},\mathcal{G}) \;\|\; f_\theta(\cdot \mid \mathcal{P},\mathcal{G})\big)\,. \tag{5}$$

(b) Null-Space Protection. At each inner step t of task i, we form an *anchor group* $a^{(i,t)}$, with its instantiations set derived from the anchor. We compute the average prediction loss $\ell_\theta(a^{(i,t)}) = \frac{1}{|a^{(i,t)}|}\sum_{d\in a^{(i,t)}} \ell_\theta(d)$ and the gradient is $g_{i,t} = \nabla_\theta \ell_\theta(a^{(i,t)})$. To prevent reshaping from impairing reasoning task performance on the anchor, we define the rank-1 projector $\Pi_g(u) = \frac{\langle u,g\rangle}{\langle g,g\rangle+\varepsilon}\,g$ and the soft null-space operator $P^{(i,t)} = I - \rho\,\Pi_{g_{i,t}}$, where $\rho \in [0,1]$ controls projection strength and $\varepsilon > 0$ ensures numerical stability. The inner-loop gradients are then replaced by their projected versions:

$$\widetilde{\nabla}_\theta \mathcal{L}_{ctr}^{(i)}(\theta_i^t) = P^{(i,t)}\,\nabla_\theta \mathcal{L}_{ctr}^{(i)}(\theta_i^t), \qquad \theta_i^{t+1} = \theta_i^t - \alpha\,\widetilde{\nabla}_\theta \mathcal{L}_{ctr}^{(i)}(\theta_i^t). \tag{6}$$

When $\rho = 1$, the update is confined to the null space of $g_{i,t}$, leaving the anchor's loss unchanged to first order. While prediction preservation maintains consistency in the model's outputs, null-space protection regulates internal parameter updates, thereby preventing catastrophic drift.

LoRA-based Edit. After circuit reshaping, we obtain the reshaped parameters θ_{rsp}. To enable fair comparison, we then apply a widely used parameter-efficient editing method LoRA on the revision set $\mathcal{D}$, yielding the adapted parameters $\theta_{\text{edit}} = \min_{\theta_{rsp}} \frac{1}{|\mathcal{D}|}\sum_{(\mathcal{P},\mathcal{G},y^*)\in\mathcal{D}} \mathrm{CE}\big(f_{\theta_{rsp}}(\cdot \mid \mathcal{P},\mathcal{G}),\, y^*\big)$, With circuit reshaping, this lightweight edit is expected to achieve improved generality and locality.

4 Experimental Settings

Datasets & Metrics. We experiment on ContextHub (Hua et al., 2024) with details in Appendix A.1. We evaluate with the *Generality* and *Locality* metrics introduced in Section 2.2.

Backbone LLM. We use Qwen2.5-3B-Instruct (Yang et al., 2025) as the backbone LLM for all experiments unless otherwise noted. This model offers competitive reasoning capability at a modest parameter scale compared to larger ones, which keeps memory and inference costs manageable.

Baselines. We compare REdit to two families of approaches. *(i) Model Reforming:* **(1) BIMT** (Liu et al., 2023b) (Brain-Inspired Modular Training) encourages functional modularity for MLPs during pretraining; we adapt it to more complex LLMs to promote separable circuits for distinct reasoning patterns, followed by LoRA-based editing. *(ii) Model Editing:* **(2) LoRA** (Hu et al., 2022) applies low-rank adapters for parameter-efficient fine-tuning and is a widely used and simple baseline in knowledge editing (Wang et al., 2024c; Jiang et al., 2024); **(3) AlphaEdit** (Fang et al., 2024) augments editing with null-space protection to reduce collateral changes; **(4) ROME** (Meng et al., 2022a) locates and updates internal representations associated with targeted knowledge. We adapt each method to the PL setting for a fair comparison. All editing methods share same $5e-5$ learning rate for fair comparison except for ROME. For other implementation details, refer to Appendix E.

Table 1: Main results on ContextHub evaluated with generality and locality metrics. The best and second-best scores are highlighted in **bold** and <u>underlined</u>, respectively. *Raw* denotes the performance of the unedited LLM. For BIMT, we apply the same LoRA-based editing method as in REdit.

Dataset	Metric	Raw	BIMT	LoRA	ROME	AlphaEdit	Ours
Level 1	**Generality**	60.7 ± 2.3	<u>72.2</u> ±1.4	63.8 ± 2.9	67.8 ±3.2	67.9 ±1.9	**74.1** ± 1.6
	Locality	N/A	61.5 ±0.7	84.9 ± 1.6	<u>89.8</u> ±3.1	87.0 ±0.9	**94.3** ± 0.4
Level 2	**Generality**	53.2 ± 1.4	<u>63.6</u> ±2.9	58.4 ± 0.1	61.3 ±1.1	58.8 ±1.5	**64.8** ± 1.2
	Locality	N/A	59.4 ±4.1	91.5 ± 0.0	93.1 ±0.1	<u>93.3</u> ±0.0	**94.3** ± 0.5
Level 3	**Generality**	45.1 ± 1.6	52.6 ±0.4	50.1 ± 0.8	51.5 ±3.3	<u>54.2</u> ±0.8	**55.0** ± 1.6
	Locality	N/A	52.3 ±1.0	92.3 ± 2.8	**94.6** ±2.7	92.2 ±0.7	<u>94.4</u> ± 0.8

5 Results and Analysis

In this section, we address five research questions: **RQ1:** How does REdit compare with existing baselines? **RQ2:** What is the contribution of each component within REdit? **RQ3:** How effectively can REdit reshape circuits in LLMs? **RQ4:** To what extent does circuit reshaping transfer to unseen circuits? **RQ5:** How does REdit perform on other domains compared to baselines?

5.1 Main Results

In this section, we address **RQ1** and present our findings in Table 1. Our analysis yields several key insights: (1) REdit consistently outperforms all baselines, achieving up to at most 16.1% improvements in generality and 12.2% in locality compared to LoRA without circuit shaping, and averaging 2.0% gains over state-of-the-art methods. (2) REdit's advantage increases as task complexity decreases, though improvements persist at all difficulty levels. This reflects that simpler tasks have more tractable circuit structures amenable to targeted reshaping. (3) BIMT achieves strong generality but poor locality due to its disruption of internal mechanisms, compromising preservation of original capabilities. (4) ROME and AlphaEdit exhibit competitive locality but inferior generality. ROME's focus on middle-layer MLPs inadequately captures distributed reasoning capabilities, while AlphaEdit's constrained editing directions limit generality enhancement to preserve other knowledge.

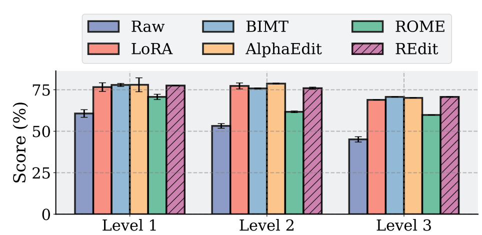


Figure 3: Editing success rates across methods on ContextHub. REdit achieves success rates comparable to other approaches, confirming that it does not compromise the model's fundamental editing capabilities.

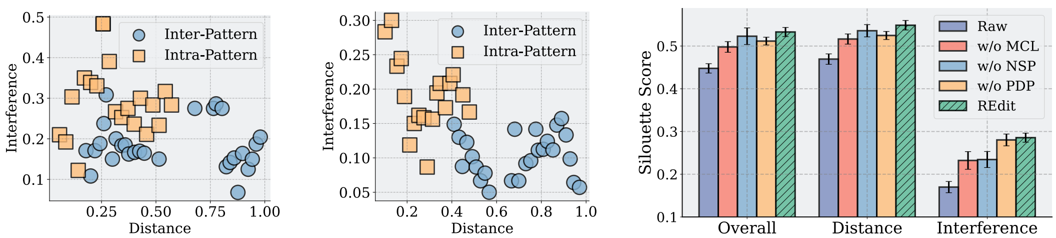
To ensure our method does not compromise the model's fundamental editing capabilities on the target instances, we evaluate editing success rates in Figure 3. Most methods achieve comparable performance, with ROME as a notable exception showing significantly lower success rates. This result further validates that restricting modifications to middle-layer MLPs is insufficient, given that reasoning capabilities in LLMs are distributed across multiple architectural components.

5.2 Additional Analysis

Ablation Study. To address **RQ2**, we conduct an ablation study with results presented in Table 2. Here, *w/o MCL* denotes the removal of Meta-Contrastive Learning, *w/o PDP* indicates without Prediction Distribution Preservation, and *w/o NSP* represents without Null Space Protection. We have the following observations: (1) All proposed components contribute meaningfully to REdit's overall performance, demonstrating their individual effectiveness. (2) Removing NSP or PDP substantially degrades performance, particularly in locality metrics, indicating that these protection mechanisms are essential for preserving model capabilities during circuit reshaping. (3) MCL provides modest but consistent improvements, attributable to enhanced optimization stability through meta-learning.

Table 2: Ablation studies on ContextHub evaluated with generality and locality metrics. The best and second-best scores are highlighted in **bold** and <u>underlined</u>, respectively.

Dataset	Metric	Raw	w/o MCL	w/o NSP	w/o PDP	**Ours**
Level 1	**Generality**	60.7 ± 2.3	72.9 ± 0.4	73.3 ± 0.2	<u>73.4</u> ± 0.5	**74.1** ± 1.6
	Locality	N/A	<u>90.7</u> ± 1.8	89.5 ± 0.3	90.1 ± 2.5	**94.3** ± 0.4
Level 2	**Generality**	53.2 ± 1.4	<u>62.5</u> ± 0.3	62.4 ± 1.6	61.3 ± 2.0	**64.8** ± 1.2
	Locality	N/A	**94.9** ± 0.6	93.0 ± 1.8	94.0 ± 0.8	<u>94.3</u> ± 0.5
Level 3	**Generality**	45.1 ± 1.6	<u>53.8</u> ± 1.3	50.9 ± 0.6	51.8 ± 0.6	**55.0** ± 1.6
	Locality	N/A	<u>93.7</u> ± 1.3	92.8 ± 1.1	92.8 ± 1.2	**94.4** ± 0.8

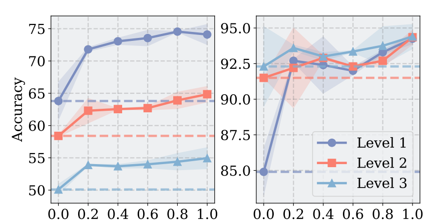


(a) Before reshaping (b) After reshaping (c) Cluster silhouette score

Figure 5: Circuit–interference relationship before and after circuit reshaping. (a,b) Scatter plots of intra- and inter-pattern measurements show improved separability in interference and circuit distance. (c) Silhouette scores across reasoning patterns indicate consistent gains in cluster separation.

Reshaping Effect on Circuit Distance. To address **RQ3**, we measure how circuit reshaping alters circuit distances between patterns. We visualize the circuit-interference relationship as described in Section 3.1, distinguishing measurements between circuits from the same reasoning pattern (Intra-Pattern) and different reasoning patterns (Inter-Pattern). Comparing the circuit-interference relationship before and after circuit reshaping in Figure 5, we observe that the two clusters become more separable in both interference and circuit distance dimensions. The right panel shows silhouette scores for the clusters across different reasoning pattern sets, where *Overall* indicates scores in the 2-dimensional space. Our results demonstrate that REdit and its components consistently improve circuit distance separation between different reasoning patterns while refining interference patterns: increasing intra-pattern interference (enhancing generality) and decreasing inter-pattern interference (improving locality). This validates both the effectiveness of our circuit reshaping approach and the Circuit-Interference Law.

Transferability of Reshaping. To address **RQ4**, we investigate the transfer of the effect of meta-contrastive circuit reshaping to unseen reasoning patterns. We apply REdit to partial reasoning patterns (20% − 80% ratio) and evaluate generality and locality on the remaining patterns. The results in Figure 4 show that while accuracy decreases slightly as the training ratio decreases, REdit consistently outperforms baselines without circuit reshaping (0% ratio) in both generality and locality metrics. This demonstrates the effectiveness of meta-contrastive learning in transferring learned circuit modifications to previously unseen reasoning patterns.



(a) Generality (b) Locality

Figure 4: Performance on unseen reasoning patterns after circuit reshaping with different ratios for training. REdit consistently outperforms baselines without reshaping.

Evaluation on Mathematics Tasks To address **RQ5**, we broaden our evaluation beyond logical tasks by assessing REdit on TemplateGSM, a mathematical reasoning benchmark. TemplateGSM encompasses multiple math templates, where each template represents a distinct reasoning pattern analogous to propositional logic reasoning patterns (detailed in Appendix A.2). The results in Figure 6 show that while all methods perform worse on TemplateGSM than on propositional logic reasoning due to the intrinsic complexity of math problems, REdit consistently outperforms all baselines, demonstrating its effectiveness on a broader range of domains. BIMT fails on both generality and locality, indicating its inability to modularize LLMs for complex tasks. Additionally, AlphaEdit and ROME show limited generality improvements, highlighting the constraints of traditional knowledge editing methods on mathematical reasoning tasks.

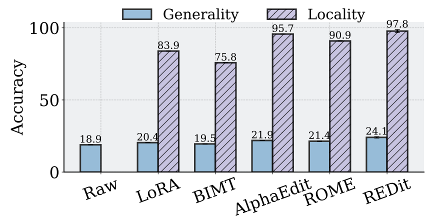


Figure 6: Evaluation on mathematical reasoning benchmark TemplateGSM.

6 Related Works

LLM Reasoning. Recent advances in LLMs have been driven significantly by their improved reasoning ability (Huang & Chang, 2022; Yu et al., 2024; Chen et al., 2025a; Li et al., 2025; Ferrag et al., 2025; Zhang et al., 2024b; Wang et al., 2024d), which is the capacity for structured, logical thinking to solve complex problems such as mathematical proofs (Ahn et al., 2024; Yang et al., 2024a), causal inference (Wang, 2024; Ma, 2024), and formal logic (Wan et al., 2024; Parmar et al., 2024). Despite their impressive performance, LLMs' reasoning abilities remain limited, especially with rigorous logical deduction (Cai et al., 2024), multi-hop inference (Yang et al., 2024b), and precise symbolic manipulation (Sullivan & Elsayed, 2024), thus prompting further improvement. Existing approaches often enhance reasoning through global strategies, such as supervised fine-tuning (Kumar et al., 2025; Zhang et al., 2025b; Luong et al., 2024) or RLHF (Hou et al., 2025; Yue et al., 2025; Wei et al., 2025). However, these methods treat reasoning as a monolithic capability rather than decomposing it into finer-grained, interpretable patterns (Havrilla et al., 2024b). As a result, they lack the precision to target and improve specific reasoning weaknesses (Chen et al., 2024). In this work, we propose a more granular reasoning editing paradigm that disentangles reasoning into distinct patterns. This enables targeted, efficient, and adaptive improvements tailored to specific reasoning challenges, moving beyond one-size-fits-all solutions.

Model Editing. Model editing modifies a pre-trained LLM's behavior post-hoc (Wang et al., 2024c), enabling error correction (Chen et al., 2025b; Li et al., 2023), knowledge updates (Wang et al., 2024a), or task adaptation without full retraining (Qi et al., 2024). Current techniques fall into several categories: memory-based methods (Liu et al., 2024b; Hu et al., 2024; Mitchell et al., 2022), meta-learning approaches (Mitchell et al., 2021; Tan et al., 2023), and localized rank-one updates (Hase et al., 2023; Meng et al., 2022a). These methods have predominantly concentrated on editing factual knowledge, typically represented as structured knowledge tuples. In contrast, reasoning editing addresses more complex reasoning processes, which are more intricately encoded within the neural circuits of LLMs (Hong et al., 2024; Kim et al., 2024). Conventional knowledge editing techniques often fail in this setting, as they struggle to satisfy the dual desiderata of generality and locality. Moreover, no prior work has systematically investigated how to directly manipulate neural circuits to enhance reasoning capabilities. In this work, we bridge this gap by taking the first step toward reasoning editing. We introduce a novel circuit-reshaping framework designed to mitigate the inherent generality–locality trade-off,thereby enabling more effective editing of reasoning patterns.

7 Conclusion

In this work, we present the first systematic study of reasoning editing, extending model editing beyond factual correction to logical inference, which introduces the generality-locality trade-off. Through circuit-level analyses, we uncover the Circuit-Interference Law, showing that interference between reasoning patterns is proportional to their circuit overlap. Inspired by this principle, we propose REdit, a framework that reshapes model circuits prior to editing to mitigate the trade-off. REdit integrates contrastive circuit shaping to align within-pattern circuits while disentangling across-pattern ones, a meta-contrastive objective to enhance generalization, and dual-level protection to preserve both prediction distributions and update directions. Empirical results show that even with a simple LoRA editor, REdit consistently outperforms knowledge editing and model reforming baselines on propositional logic across three difficulty tiers using Qwen-2.5-3B. Additional experiments further demonstrate its potential across different reasoning domains.

References

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*, 2024.

Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. Forest-of-thought: Scaling test-time compute for enhancing llm reasoning. *arXiv preprint arXiv:2412.09078*, 2024.

Chengkun Cai, Xu Zhao, Haoliang Liu, Zhongyu Jiang, Tianfang Zhang, Zongkai Wu, Jenq-Neng Hwang, Serge Belongie, and Lei Li. The role of deductive and inductive reasoning in large language models. *arXiv preprint arXiv:2410.02892*, 2024.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025a.

Qizhou Chen, Taolin Zhang, Chengyu Wang, Xiaofeng He, Dakan Wang, and Tingting Liu. Attribution analysis meets model editing: Advancing knowledge correction in vision language models with visedit. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 2168–2176, 2025b.

Yulong Chen, Yang Liu, Jianhao Yan, Xuefeng Bai, Ming Zhong, Yinghao Yang, Ziyi Yang, Chenguang Zhu, and Yue Zhang. See what llms cannot answer: A self-challenge framework for uncovering llm weaknesses. *arXiv preprint arXiv:2408.08978*, 2024.

Inyoung Cheong, King Xia, KJ Kevin Feng, Quan Ze Chen, and Amy X Zhang. (a) i am not a lawyer, but...: engaging legal experts towards responsible llm policies for legal advice. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pp. 2454–2469, 2024.

Yingqian Cui, Pengfei He, Jingying Zeng, Hui Liu, Xianfeng Tang, Zhenwei Dai, Yan Han, Chen Luo, Jing Huang, Zhen Li, et al. Stepwise perplexity-guided refinement for efficient chain-of-thought reasoning in large language models. *arXiv preprint arXiv:2502.13260*, 2025.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*, 2021.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Shi Jie, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. *arXiv preprint arXiv:2410.02355*, 2024.

Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. From llm reasoning to autonomous ai agents: A comprehensive review. *arXiv preprint arXiv:2504.19678*, 2025.

Xiou Ge, Ali Mousavi, Edouard Grave, Armand Joulin, Kun Qian, Benjamin Han, Mostafa Arefiyan, and Yunyao Li. Time sensitive knowledge editing through efficient finetuning. *arXiv preprint arXiv:2406.04496*, 2024.

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*, 2023.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in Neural Information Processing Systems*, 36:17643–17668, 2023.

Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. Teaching large language models to reason with reinforcement learning. *arXiv preprint arXiv:2403.04642*, 2024a.

Alex Havrilla, Sharath Raparthy, Christoforus Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, and Roberta Raileanu. Glore: When, where, and how to improve llm reasoning via global and local refinements. *arXiv preprint arXiv:2402.10963*, 2024b.

Yinhan He, Wendy Zheng, Yushun Dong, Yaochen Zhu, Chen Chen, and Jundong Li. Towards global-level mechanistic interpretability: A perspective of modular circuits of large language models. In *Forty-second International Conference on Machine Learning*.

Guan Zhe Hong, Nishanth Dikkala, Enming Luo, Cyrus Rashtchian, Xin Wang, and Rina Panigrahy. How transformers solve propositional logic problems: A mechanistic analysis. *arXiv preprint arXiv:2411.04105*, 2024.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Wilke: Wise-layer knowledge editor for lifelong knowledge editing. *arXiv preprint arXiv:2402.10987*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Wenyue Hua, Kaijie Zhu, Lingyao Li, Lizhou Fan, Shuhang Lin, Mingyu Jin, Haochen Xue, Zelong Li, JinDong Wang, and Yongfeng Zhang. Disentangling logic: The role of context in large language model reasoning capabilities. *arXiv preprint arXiv:2406.02787*, 2024.

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*, 2022.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

Gangwei Jiang, Yahui Liu, Zhaoyi Li, Qi Wang, Fuzheng Zhang, Linqi Song, Ying Wei, and Defu Lian. What makes a good reasoning chain? uncovering structural patterns in long chain-of-thought reasoning. *arXiv preprint arXiv:2505.22148*, 2025.

Yuxin Jiang, Yufei Wang, Chuhan Wu, Wanjun Zhong, Xingshan Zeng, Jiahui Gao, Liangyou Li, Xin Jiang, Lifeng Shang, Ruiming Tang, et al. Learning to edit: Aligning llms with knowledge editing. *arXiv preprint arXiv:2402.11905*, 2024.

Geonhee Kim, Marco Valentino, and André Freitas. A mechanistic interpretation of syllogistic reasoning in auto-regressive language models. *arXiv preprint arXiv:2408.08590*, 2024.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. Llm post-training: A deep dive into reasoning large language models. *arXiv preprint arXiv:2502.21321*, 2025.

Yuxiang Lai, Jike Zhong, Ming Li, Shitian Zhao, and Xiaofeng Yang. Med-r1: Reinforcement learning for generalizable medical reasoning in vision-language models. *arXiv preprint arXiv:2503.13939*, 2025.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*, 2025.

Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. Unveiling the pitfalls of knowledge editing for large language models. *arXiv preprint arXiv:2310.02129*, 2023.

Yuanze Lin, Xun Guo, and Yan Lu. Self-supervised video representation learning with meta-contrastive network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8239–8249, 2021.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Jiateng Liu, Pengfei Yu, Yuji Zhang, Sha Li, Zixuan Zhang, and Heng Ji. Evedit: Event-based knowledge editing with deductive editing boundaries. *arXiv preprint arXiv:2402.11324*, 2024b.

Wentao Liu, Hanglei Hu, Jie Zhou, Yuyang Ding, Junsong Li, Jiayi Zeng, Mengliang He, Qin Chen, Bo Jiang, Aimin Zhou, et al. Mathematical language models: A survey. *arXiv preprint arXiv:2312.07622*, 2023a.

Ziming Liu, Eric Gan, and Max Tegmark. Seeing is believing: Brain-inspired modular training for mechanistic interpretability. *Entropy*, 26(1):41, 2023b.

Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. Are emergent abilities in large language models just in-context learning? *arXiv preprint arXiv:2309.01809*, 2023.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, et al. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2024.

Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. Reft: Reasoning with reinforced fine-tuning. *arXiv preprint arXiv:2401.08967*, 3, 2024.

Jing Ma. Causal inference with large language model: A survey. *arXiv preprint arXiv:2409.09822*, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022a.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022b.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, pp. 15817–15831. PMLR, 2022.

Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. Logicbench: Towards systematic evaluation of logical reasoning ability of large language models. *arXiv preprint arXiv:2404.15522*, 2024.

Gabrijela Perković, Antun Drobnjak, and Ivica Botički. Hallucinations in llms: Understanding and addressing challenges. In *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, pp. 2084–2088. IEEE, 2024.

Siyuan Qi, Bangcheng Yang, Kailin Jiang, Xiaobo Wang, Jiaqi Li, Yifan Zhong, Yaodong Yang, and Zilong Zheng. In-context editing: Learning knowledge from self-induced distributions. *arXiv preprint arXiv:2406.11194*, 2024.

Raimundo Real and Juan M Vargas. The probabilistic basis of jaccard's index of similarity. *Systematic biology*, 45(3):380–385, 1996.

Jintian Shao and Yiming Cheng. Cot is not true reasoning, it is just a tight constraint to imitate: A theory perspective. *arXiv preprint arXiv:2506.02878*, 2025.

Shamus Sim and Tyrone Chen. Critique of impure reason: Unveiling the reasoning behaviour of medical large language models. *arXiv preprint arXiv:2412.15748*, 2024.

Rob Sullivan and Nelly Elsayed. Can large language models act as symbolic reasoners? *arXiv preprint arXiv:2410.21490*, 2024.

Alan Sun. Circuit stability characterizes language model generalization. *arXiv preprint arXiv:2505.24731*, 2025.

Zhongxiang Sun. A short survey of viewing large language models in legal aspect. *arXiv preprint arXiv:2303.09136*, 2023.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. *arXiv preprint arXiv:2310.10348*, 2023.

Chenmien Tan, Ge Zhang, and Jie Fu. Massive editing for large language models via meta learning. *arXiv preprint arXiv:2311.04661*, 2023.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Will we run out of data? limits of llm scaling based on human-generated data. *arXiv preprint arXiv:2211.04325*, 2022.

Yuxuan Wan, Wenxuan Wang, Yiliu Yang, Youliang Yuan, Jen-tse Huang, Pinjia He, Wenxiang Jiao, and Michael R Lyu. Logicasker: Evaluating and improving the logical reasoning ability of large language models. *arXiv preprint arXiv:2401.00757*, 2024.

Boshi Wang, Xiang Yue, and Huan Sun. Can chatgpt defend its belief in truth? evaluating llm reasoning via debate. *arXiv preprint arXiv:2305.13160*, 2023.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *Advances in Neural Information Processing Systems*, 37:53764–53797, 2024a.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. Rethinking the bounds of llm reasoning: Are multi-agent discussions the key? *arXiv preprint arXiv:2402.18272*, 2024b.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey. *ACM Computing Surveys*, 57(3):1–37, 2024c.

Yiqi Wang, Wentao Chen, Xiaotian Han, Xudong Lin, Haiteng Zhao, Yongfei Liu, Bohan Zhai, Jianbo Yuan, Quanzeng You, and Hongxia Yang. Exploring the reasoning abilities of multimodal large language models (mllms): A comprehensive survey on emerging trends in multimodal reasoning. *arXiv preprint arXiv:2401.06805*, 2024d.

Zeyu Wang. Causalbench: A comprehensive benchmark for evaluating causal reasoning capabilities of large language models. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pp. 143–151, 2024.

Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I Wang. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution. *arXiv preprint arXiv:2502.18449*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Kaiyu Yang, Gabriel Poesia, Jingxuan He, Wenda Li, Kristin Lauter, Swarat Chaudhuri, and Dawn Song. Formal mathematical reasoning: A new frontier in ai. *arXiv preprint arXiv:2412.16075*, 2024a.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? *arXiv preprint arXiv:2402.16837*, 2024b.

Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. Knowledge circuits in pretrained transformers. *arXiv preprint arXiv:2405.17969*, 2024.

Fei Yu, Hongbo Zhang, Prayag Tiwari, and Benyou Wang. Natural language reasoning, a survey. *ACM Computing Surveys*, 56(12):1–39, 2024.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095, 2007.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. When scaling meets llm finetuning: The effect of data, model and finetuning method. *arXiv preprint arXiv:2402.17193*, 2024a.

Lin Zhang, Lijie Hu, and Di Wang. Mechanistic unveiling of transformer circuits: Self-influence as a key to model reasoning. *arXiv preprint arXiv:2502.09022*, 2025a.

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E Gonzalez. Tempera: Test-time prompting via reinforcement learning. *arXiv preprint arXiv:2211.11890*, 2022.

Xinlu Zhang, Zhiyu Zoey Chen, Xi Ye, Xianjun Yang, Lichang Chen, William Yang Wang, and Linda Ruth Petzold. Unveiling the impact of coding data instruction fine-tuning on large language models reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25949–25957, 2025b.

Yadong Zhang, Shaoguang Mao, Tao Ge, Xun Wang, Adrian de Wynter, Yan Xia, Wenshan Wu, Ting Song, Man Lan, and Furu Wei. Llm as a mastermind: A survey of strategic reasoning with large language models. *arXiv preprint arXiv:2404.01230*, 2024b.

Yifan Zhang. Training and evaluating language models with template-based data generation. *arXiv preprint arXiv:2411.18104*, 2024.

Yufeng Zhang, Xuepeng Wang, Lingxiang Wu, and Jinqiao Wang. Enhancing chain of thought prompting in large language models via reasoning patterns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25985–25993, 2025c.

Zhuoran Zhang, Yongxiang Li, Zijian Kan, Keyuan Cheng, Lijie Hu, and Di Wang. Locate-then-edit for multi-hop factual recall under knowledge editing. *arXiv preprint arXiv:2410.06331*, 2024c.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. Dyval: Dynamic evaluation of large language models for reasoning tasks. *arXiv preprint arXiv:2309.17167*, 2023.

Ethics & Reproducibility Statement

We use only public, anonymized datasets. No human subjects or sensitive data are involved. The work aims to improve LLM reliability, aligning with the ICLR Code of Ethics.

For reproducibility, datasets, settings, and metrics are detailed in the paper and appendix. Code and instructions are released in the anonymous repository.

The Use of Large Language Models

In this study, we mainly leverage LLMs to enhance the clarity and polish of the manuscript. All conceptual development and methodology design were conducted by the authors.

A Dataset Details

A.1 Propositional Logic: ContextHub

ContextHub (Hua et al., 2024) is a benchmark for propositional logical reasoning, built on top of formal logic templates generated by DyVal (Zhu et al., 2023). It dynamically instantiated these templates into natural language questions across 11 real-world domains drawn from Wikipedia (e.g., culture, health, technology) along with an abstract form, thereby ensuring both diversity and robustness of reasoning scenarios.

Statistics. ContextHub consists of a total of 256 formal logic templates, spanning several difficulty levels. Each template is instantiated across 12 domains with 5 variations per domain. This yields 360 samples for level-1 logic, 600 for level-2 logic, and 2,880 for level-3 logic types. Each sample is balanced across the three answer labels (`True`, `False`, `N/A`). In this work, we treat each logic template as a distinct reasoning pattern.

Example. Table 3 illustrates both an abstract and a contextual instantiation of the same level-1 template. The abstract form substitutes propositional variables with arbitrary character sequences, while the contextual form grounds them in a concrete domain.

Abstract Instance	**Contextual Instance**
$(\texttt{vxkgr} \vee \texttt{caunc}) \rightarrow \texttt{ybyz}$. Given `ybyz` is False, what is the value of `caunc`?	If an area of land has experienced significant uplift or been shaped by powerful erosional forces, then the terrain will feature tall, steep mountains. Given that the area does not have tall, steep mountains, can it be determined if powerful erosional forces have shaped the land?

Table 3: Level-1 example instantiations in ContextHub.

A.2 Mathematics: TemplateGSM

TemplateGSM (Zhang, 2024) is a large-scale benchmark for mathematical reasoning, constructed using the Template-based Data Generation (TDG) paradigm. Frontier LLMs (e.g., GPT-4) are employed to author parameterized meta-templates, which are then instantiated into natural language problems paired with programmatically verifiable solutions. This ensures not only linguistic and structural diversity but also guarantees correctness at scale.

Statistics. TemplateGSM comprises 7,473 GPT-4-authored templates, instantiated into approximately 7.47 million grade-school math problems spanning arithmetic, fractions, percentages, and elementary algebra. Problem lengths range from 18–636 tokens. In this work, we experiment on a curated subset of 600 problems, each restricted to a single numerical answer (integer or float).

Example. Table 4 illustrates a GPT-4-authored template alongside one instantiated problem, highlighting how TDG generates diverse mathematical reasoning tasks.

Math Template	**Instantiated Problem**
`[NAME]` sold `[NUM1]` `[ITEM]` to `[her/his/their]` friends in April at a `[LOCATION]` in `[COUNTY]`, `[STATE]`. In May, `[PRONOUN]` sold `[NUM2]` `[ITEM]`. How many `[ITEM]` did `[NAME]` sell altogether in April and May?	Rosy Plascencia sold 238 air fryers to her friends in April at a yoga studio boutique in Bracken County, Kentucky. In May, they sold 119 air fryers. How many air fryers did Rosy Plascencia sell altogether in April and May?

Table 4: Example instantiations in TemplateGSM.

B Circuit Distance Metric

Given two patterns π_i, π_j and their *attributed circuits* as the sets of top–τ edges ranked by attribution scores: $\mathcal{C}_\pi^{(\tau)} = \{(e, w_\pi(e)) : \ w_\pi(e) \geq t_\pi(\tau)\}$, we quantify structural dissimilarity between π_i and π_j using three complementary metrics.

(a) Weighted Jaccard Distance (Real & Vargas, 1996).

$$d_{\mathrm{Jac}}(i,j) \ = \ 1 \ - \ \frac{\sum_{e \in \mathcal{C}_{\pi_i}^{(\tau)} \cup \mathcal{C}_{\pi_j}^{(\tau)}} \min\{w_i(e),\, w_j(e)\}}{\sum_{e \in \mathcal{C}_{\pi_i}^{(\tau)} \cup \mathcal{C}_{\pi_j}^{(\tau)}} \max\{w_i(e),\, w_j(e)\} + \varepsilon}. \tag{7}$$

This emphasizes overlap of influential edges in the two attributed circuits.

(b) Edit Distance (Yujian & Bo, 2007).

$$d_{\mathrm{Edit}}(i,j) \ = \ \frac{\sum_{e \in \mathcal{C}_{\pi_i}^{(\tau)} \cup \mathcal{C}_{\pi_j}^{(\tau)}} |\, w_i(e) - w_j(e)\, |}{\sum_{e \in \mathcal{C}_{\pi_i}^{(\tau)} \cup \mathcal{C}_{\pi_j}^{(\tau)}} \max\{w_i(e),\, w_j(e)\} + \varepsilon}. \tag{8}$$

This captures the minimal "edit cost" required to reconcile the two circuits.

(c) Optimal-Transport (OT) Distance (Cuturi, 2013). Normalize edge weights in each attributed circuit to probability masses

$$p_\pi(e) = \frac{w_\pi(e)}{\sum_{e' \in \mathcal{C}_\pi^{(\tau)}} w_\pi(e')}, \quad e \in \mathcal{C}_\pi^{(\tau)}.$$

Let $c(e, e') \geq 0$ denote a ground cost between edges (e.g., based on layer/head/type and token-span offsets). The optimal transport distance is then

$$\begin{aligned} d_{\mathrm{OT}}(i,j) \ &= \ \min_{T \in \Pi(p_i, p_j)} \sum_{e \in \mathcal{C}_{\pi_i}^{(\tau)}} \sum_{e' \in \mathcal{C}_{\pi_j}^{(\tau)}} T_{e,e'}\, c(e, e'), \\ \Pi(p_i, p_j) &= \{T \geq 0 : \sum_{e'} T_{e,e'} = p_i(e),\, \sum_{e} T_{e,e'} = p_j(e')\}. \end{aligned} \tag{9}$$

This explicitly accounts for circuit geometry by measuring the minimal mass transport needed to align the two attributed circuits.

C Bonus Effect of REdit

In this section, we compare the performance of the original LLMs with that of the unedited models after undergoing REdit circuit reshaping in Figure 7. Surprisingly, we observe that even without explicit editing, REdit consistently yields modest accuracy gains across three difficulty levels of logical reasoning tasks, with the largest improvements occurring on the easier problems. We attribute this phenomenon to circuit reshaping's ability to reorganize the model's internal mechanisms, where it might suppresses noisy or erroneous circuits while preserving task-critical ones, thereby enhancing the model's overall reasoning performance. We will explore this phenomenon further in the future.

D Algorithm

In this section, we provide the algorithm of REdit circuit reshaping in Algorithm 1.

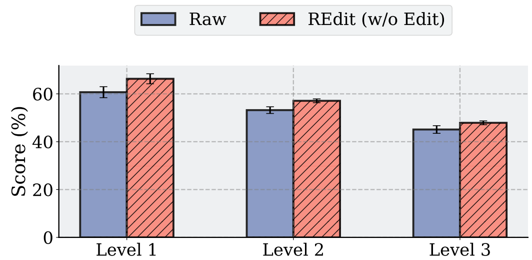


Figure 7: Comparison of accuracy between the original LLMs (*Raw*) and the unedited models after REdit circuit reshaping (*REdit (w/o Edit)*) across three difficulty levels of logical reasoning tasks. REdit consistently provides modest gains, with the most notable improvements at Level 1.

Algorithm 1 REdit Circuit Reshaping

procedure REDIT($\theta, \Pi_{\text{train}}, \eta, \alpha, s, \rho$)
  Input: LLM θ; training patterns Π_{train}; rates η, α; steps s; ratio ρ.
  Output: Reshaped LLM θ'
  Contrastive Circuit Shaping:
  Derive attribution scores $\tilde{w}_\pi = w_\pi / \|w_\pi\|_2$; define InfoNCE loss $\mathcal{L}_{\text{ctr}}(\theta)$ as in Eq. equation 3
  Meta-Contrastive Learning with Dual Protection:
  for each meta-iteration **do**
    Sample batch $\mathcal{B} \subset \Pi_{\text{train}}$
    for each $i \in \mathcal{B}$ **do** ▷ Inner loop (equation 4)
      Initialize $\theta_i^0 \leftarrow \theta$
      for $t = 0, 1, \ldots, s-1$ **do**
        Compute preservation loss $\mathcal{L}_{\text{pred}}(\theta_i^t)$ as in Eq. equation 5
        Inner objective: $\mathcal{L}_{\text{inner}}^{(i)} = \mathcal{L}_{\text{ctr}}^{(i)} + \lambda \mathcal{L}_{\text{pred}}$
        Derive gradient of inner objective: $g_{i,t} \leftarrow \nabla_\theta \mathcal{L}_{\text{inner}}^{(i)}(\theta_i^t)$
        Form projector $P^{(i,t)} = I - \rho \Pi_{g_{i,t}}$ ▷ Null-space protection
        Update $\theta_i^{t+1} \leftarrow \theta_i^t - \alpha P^{(i,t)} g_{i,t}$ ▷ Protected update (6)
      end for
      Set $\phi_i \leftarrow \theta_i^s$
    end for
    Outer update: $\theta \leftarrow \theta + \eta \cdot \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} (\phi_i - \theta)$ ▷ Meta update, Eq. equation 4
  end for
  return θ_{REdit}
end procedure

E IMPLEMENTATION DETAILS

For *circuit reshaping*, we set the inner learning rate to $\alpha = 1 \times 10^{-6}$ and the outer learning rate to $\eta = 1 \times 10^{-6}$, running for 200 steps with an inner update step size of $s = 5$. In each iteration, we sample $|\mathcal{B}| = 2$ contrastive pairs of reasoning patterns in a batch. The temperature for *contrastive circuit shaping* is fixed at $\tau_t = 1$. The *null-space protection* coefficient is set to $\rho = 0.5$, and the *prediction distribution preservation* weight is $\lambda = 0.1$. When computing attribution scores, we use $K = 10$ instantiations for circuit distance calculation and $K = 2$ instantiations for REdit circuit reshaping due to computational restricts. For experiments validating Circuit-Interference Law, we construct circuits with top-$\tau = 5\%$ edges. During editing, we modify one instance per sample. For *LoRA-based editing*, we use a learning rate of 5×10^{-5} for 10 steps. Unless otherwise specified, the same learning rate of 5×10^{-5} is adopted for other baselines to ensure fair comparison. All experiments are conducted on four A100 GPUs; each REdit meta-iteration consumes ≈ 1 minute.

Corrupt Dataset. To construct the corrupt dataset, we modify the final question to query the status of the first propositional variable in the premise $\mathcal{P}$ (fact-checking), instead of the status of the goal $\mathcal{G}$, while keeping all other components unchanged.

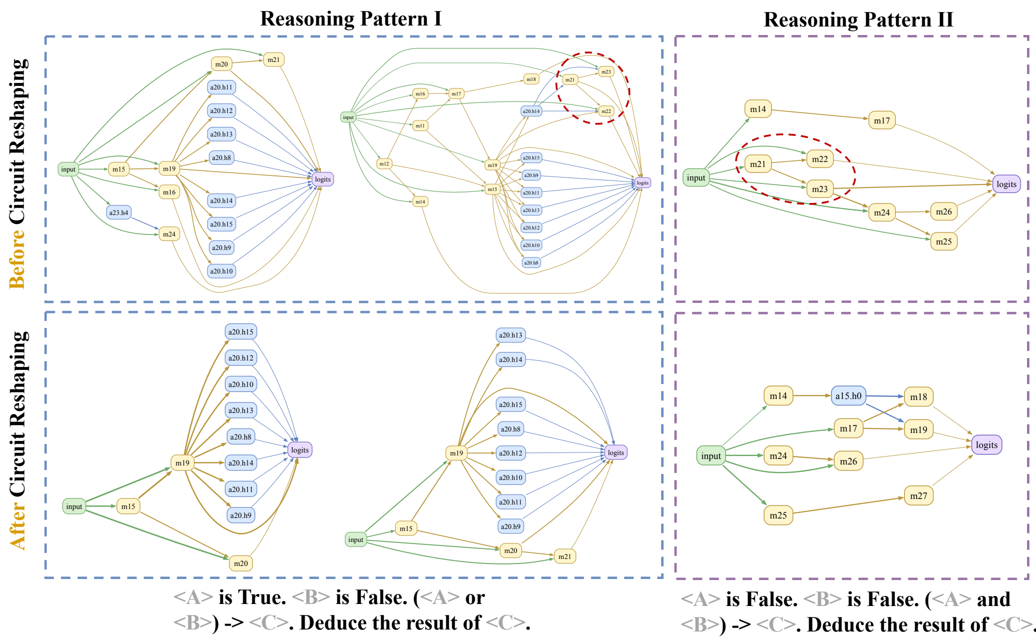


Figure 8: Case study of circuits from reasoning patterns **I** and **II** before and after REdit circuit reshaping. REdit enhances intra-pattern consistency while eliminating inter-pattern overlap.

Prompts. For the propositional logic dataset, we append the instruction: `(Answer only in True, False, or N/A (Neither)). Answer:` to each question. For the mathematical dataset, we append: `Answer with only the final numeric result. Answer:` to ensure precise and standardized responses.

F Case Study

In this section, we present a case study illustrating the circuits of two reasoning patterns before and after REdit circuit reshaping. As shown in Figure 8, prior to reshaping, circuits from different instantiations of reasoning pattern **I** exhibit substantial overlap, though discrepancies remain, most notably around node `a23.h4` and the tree structure formed by `m21`, `m22`, and `m23`. Circuits from reasoning pattern **II** share slight overlap with those of pattern **I**, particularly within the same tree structure. After circuit reshaping, circuits from different instantiations of reasoning pattern **I** become more consistent and exhibit stronger alignment, with noisy nodes and edges effectively pruned. At the same time, overlap between circuits of reasoning patterns **I** and **II** is almost completely eliminated. This case study highlights the effectiveness of REdit: it reshapes circuits to achieve greater separation across different reasoning patterns while producing more coherent and centralized structures within the same reasoning pattern.

G Generality-Locality Trade-Off of REdit

In this section, we compare the generality–locality trade-off before and after applying circuit reshaping with REdit. As shown in Figure 9, across different learning rates, LLMs trained with REdit consistently achieve a superior Pareto frontier compared to raw LLMs, highlighting the effectiveness of our approach.

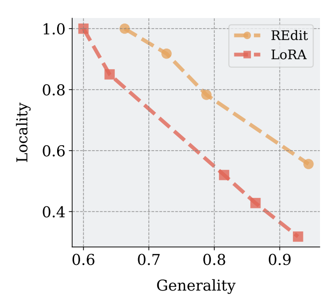


Figure 9: Trade-off of REdit