RESCUE: REPAIRING LANGUAGE MODEL ERRORS TO SPARSE CIRCUITS VIA REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) exhibit strong general capabilities that mechanistic interpretability has attributed to sparse computational circuits. However, existing circuit studies emphasize preserving functionality or explaining safety, leaving the mechanisms underlying failures across a broader range of tasks largely unexplored. Extending circuit analysis from abilities to errors, we explore the perspective that such failures may likewise arise from erroneous internal computations and that targeted tuning of the corresponding parameters can correct such errors while largely preserving other capabilities. Motivated by this insight, we introduce **RESCUE** (**R**easoning-**E**rror **S**parse-**C**ircuit **U**ncovering and **E**diting), a framework that localizes error-associated circuits and surgically repairs them for performance enhancement. General tasks typically involve multi-step reasoning and long-form generation, where early deviations can cause prefixes to drift from supervised references, leading SFT-based mask optimization to overlook circuits involved in generation-time errors. RESCUE therefore refines these masks through reinforcement learning with multiple masked-model rollouts, improving their relevance to observed task failures. Finally, RESCUE introduces a pruning technique and precisely fine-tunes error circuits to correct task failures, thereby translating error localization into a sparse and targeted model update. We validate RESCUE on heterogeneous repair sets across two domains: **(1) mathematical reasoning**, identifying a math error circuit of 1.40% density whose repair raises accuracy from $6.0\% \rightarrow 75.5\%$; and **(2) medical QA**, where a similarly compact 1.44% circuit improves repair-set accuracy from $0\% \rightarrow 81\%$. Our code is available at: https://anonymous.4open.science/r/RESCUE-1F46/.

1 INTRODUCTION

Recent advances (Ouyang et al., 2022; Rafailov et al., 2023) have enabled large language models (LLMs) to achieve strong performance across a broad range of tasks. Nevertheless, they continue to exhibit recurring failures even on seemingly simple problems, such as elementary arithmetic, limiting their reliability (Zhang et al., 2024b; Nikankin et al., 2025; Yu et al., 2025a). Existing analyses and remedies often emphasize data coverage or training objectives (Lin et al., 2022; Kandpal et al., 2023; Ji et al., 2023), but provide limited insight into the internal computations responsible for these errors. Meanwhile, growing evidence suggests that specific model behaviors are mediated by sparse internal components and computational circuits (Dai et al., 2022; Todd et al., 2024; Meng et al., 2022). Motivated by these findings, we propose that recurring failures may likewise arise from faulty computations concentrated in sparse error-associated circuits, and that localizing them enables effective repair through circuit-restricted tuning with limited impact on other capabilities.

We categorize existing studies related to error attribution into two main research lines: **(I) Task circuits.** Prior work has uncovered sparse circuits underlying capabilities such as factual recall and reasoning (Yao et al., 2024; Hong et al., 2025; Hou et al., 2023), but has primarily sought circuits that faithfully reproduce or preserve successful task behavior. Their task-specific attribution methods do not readily transfer to error attribution or targeted repair. **(II) Trustworthy circuits.** Safety-oriented research demonstrates that localized components can support selective interventions on alignment (Yu et al., 2026) and backdoor behaviors (Zhao et al., 2025; Yu et al., 2025b), yet remain focused on safety-specific mechanisms. Hallucination-attribution work such as H-Neurons (Gao et al., 2025) links erroneous outputs to individual neurons. However, whether localized interventions

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.

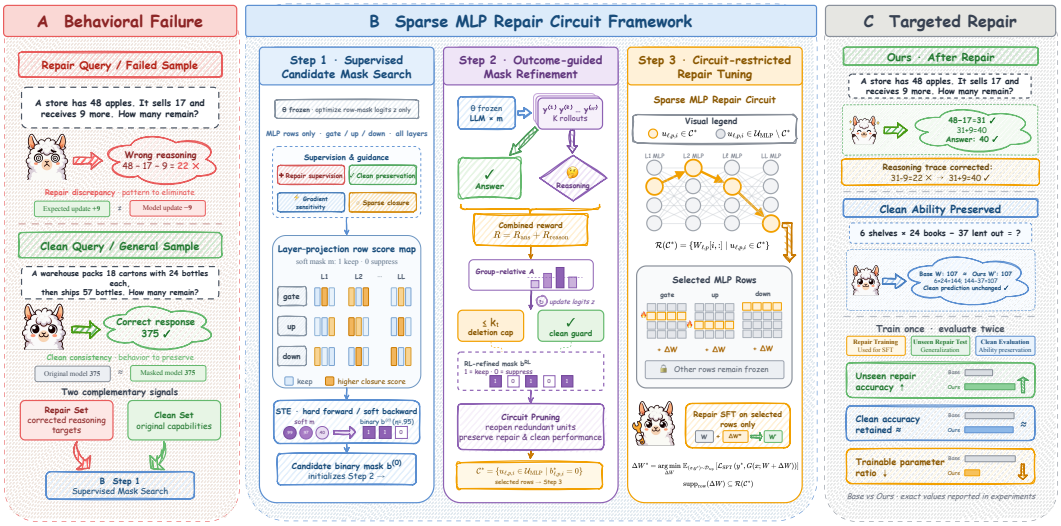


Figure 1: Overview of **RESCUE**, with complementary repair and preservation signals (***Left***), the sparse-circuit localization and tuning framework (***Middle***), and targeted repair generations (***Right***).

can repair a wider range of failures remains unexplored. Together, there is a lack of methods that can efficiently locate circuits and use them for targeted repair across broader task domains.

To bridge these gaps, we propose **RESCUE** (**R**easoning-**E**rror **S**parse-**C**ircuit **U**ncovering and **E**diting), a framework that localizes sparse circuits associated with task failures and selectively tunes them to improve the corresponding task performance. RESCUE localizes sparse circuits associated with model failures through a two-stage optimization. We first isolate error-associated circuits through SFT-based mask optimization using teacher-corrected responses, with model weights frozen. Yet the mismatch between offline references and free generation may prevent mask optimization from identifying all components contributing to erroneous outputs. To improve the accuracy and generalizability of error-circuit localization, we refine the candidate masks in the second stage using feedback from multiple rollouts of the masked model. With a group-relative policy-gradient objective over these multiple generations, RESCUE updates the mask toward components that consistently contribute to failure. The refined masks are then pruned to obtain the final sparse circuit, and only the corresponding parameters are tuned for targeted repair.

We evaluate RESCUE on Qwen3-8B (Yang et al., 2025) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024). In the mathematical domain, RESCUE identifies sparse circuits associated with model errors in both models, and then fine-tuning only these localized parameters raises GSM8K (Cobbe et al., 2021) accuracy from $80\% \rightarrow 91\%$ and improves MATH-500 (Hendrycks et al., 2021b; Lightman et al., 2024) from $75\% \rightarrow 78\%$. RESCUE outperforms the supervised-only interpretable baseline and vanilla fine-tuning on GSM8K by 4 and 3 percentage points, respectively, highlighting the benefit of refining masks according to response-level task performance. In the medical domain, RESCUE likewise outperforms vanilla fine-tuning on the MedMCQA (Pal et al., 2022) repair sets. Across all repair settings, RESCUE retains an average of 99.4% of baseline aggregate performance across eight non-target benchmarks, indicating that targeted repair largely preserves general capabilities. These results demonstrate that attributing task failures to sparse internal mechanisms provides an effective basis for targeted repair while largely preserving non-target capabilities.

In summary, we introduce RESCUE, a generation-guided framework that combines supervised mask optimization, reinforcement learning over multiple masked-model rollouts, and circuit pruning to localize sparse circuits associated with task failures. Their causal effects transfer across input rephrasings, and tuning only the corresponding parameters enables targeted error correction while largely preserving non-target performance, illustrating how circuit attribution can support model repair.

2 Related Work

Circuit Discovery. Circuit discovery explains model behavior through sparse computational subgraphs (Wang et al., 2023). Methods span Path Patching and ACDC (Goldowsky-Dill et al., 2023;

Conmy et al., 2023), scalable Attribution Patching, Edge Pruning, and IBCircuit (Syed et al., 2024; Bhaskar et al., 2024; Bian et al., 2025), and feature-level Sparse Feature Circuits and CircuitTracer (Marks et al., 2025; Hanna et al., 2025), with applications to performance improvement (Yao et al., 2024; Cho et al., 2025; Kharlapenko et al., 2025). Yet heuristic search requires costly repeated interventions, while scalable alternatives target behavior preservation rather than error attribution. RESCUE instead combines supervised mask optimization with feedback from multiple rollouts, avoiding exhaustive search and enabling generation-aware error localization for targeted repair.

Error Attribution. Hidden-state methods probe latent knowledge with CCS (Burns et al., 2023), falsehood and hallucination (Azaria & Mitchell, 2023; Chen et al., 2024), truthfulness (Orgad et al., 2025), and arithmetic errors (Sun et al., 2025a), but decodability does not establish causality. Component-level work attributes factual errors to attention and MLP mechanisms through SAT Probe (Yuksekgonul et al., 2024), mechanistic tracing (Yu et al., 2024), FAITH (Yuan et al., 2024), and ReDeEP (Sun et al., 2025b), while arithmetic studies identify critical heads, MLPs, and circuits (Zhang et al., 2024b; Hanna et al., 2023). ITI (Li et al., 2023), TruthX (Zhang et al., 2024a), and DoLa (Chuang et al., 2024) further use internal signals for inference-time intervention.

3 PRELIMINARY

LLM Computational Graph. Let M_θ denote a language model parameterized by θ. Following the computational-graph view of language models (Yao et al., 2024), we represent its forward computation as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. At a chosen level of abstraction, each node $v \in \mathcal{V}$ represents an internal computation, while each directed edge $(u, v) \in \mathcal{E}$ specifies how the output of an upstream computation u contributes to a downstream computation v. Given an input sequence $\mathbf{x}$, the complete graph induces the conditional output distribution $p_\theta(\mathbf{y} \mid \mathbf{x}; \mathcal{G})$.

Circuit Units. We use an output-dimension-level granularity for Transformer MLP projections, following prior circuit-attribution work (Geva et al., 2022; Yu et al., 2026). Specifically, each output dimension of the gate, up, and down projections is treated as an individual circuit unit. For projection $p \in \{\text{gate}, \text{up}, \text{down}\}$ in layer l, let $\mathbf{W}_{l,p} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ denote its weight matrix. Given an input activation $\mathbf{h} \in \mathbb{R}^{d_{\text{in}}}$, the j-th unit produces the corresponding scalar projection output as follows:

$$n_{l,p,j}(\mathbf{h}) = [\mathbf{W}_{l,p}\mathbf{h}]_j = \mathbf{w}_{l,p,j}^\top \mathbf{h},$$

where $\mathbf{w}_{l,p,j}^\top$ is the j-th row of $\mathbf{W}_{l,p}$. We index each unit as $u = (l, p, j)$ and denote the set of all such units across MLP projections and Transformer layers by $\mathcal{U}$.

Circuit Formulation. For a task distribution $\mathcal{D}$, a circuit $\mathcal{C} \subseteq \mathcal{G}$ is a sparse computational subgraph that faithfully retains the task-relevant behavior of the complete model (Conmy et al., 2023; Bhaskar et al., 2024). Circuit discovery can therefore be formulated as

$$\min_{\mathcal{C} \subseteq \mathcal{G}} \quad \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[D_{\text{KL}}(p_\theta(\cdot \mid \mathbf{x}; \mathcal{G}) \,\|\, p_\theta(\cdot \mid \mathbf{x}; \mathcal{C})) \right] \text{ s.t. } \quad \frac{|\mathcal{C}|}{|\mathcal{G}|} \leq \gamma, \tag{1}$$

where D_{KL} measures the behavioral discrepancy between the circuit and the complete model, and γ specifies the upper bound on the relative circuit size. Unlike conventional circuits defined as retained subgraphs, RESCUE defines an error-associated circuit as a sparse set of units whose suppression corrects model failures while preserving behavior on clean inputs. Here, $|\cdot|$ counts computational elements under the granularity adopted by the circuit-discovery method.

4 RESCUE

We present **RESCUE**, a framework for localizing and repairing sparse error-associated circuits in general tasks. We first construct complementary error-correction and clean datasets (Sec. 4.1), then localize circuits through supervised mask search, output-guided reinforcement learning, and pruning (Sec. 4.2), before circuit validation and circuit-restricted tuning (Sec. 4.3).

4.1 DATA CONSTRUCTION

To construct dual-path supervision for error-circuit discovery, we first run the target model on each question x and partition the resulting responses according to final-answer correctness. For a cor-

rectly answered question x_j, we retain the model's original response y_j, forming the clean set $\mathcal{D}_{\text{clean}} = \{(x_j, y_j)\}$. For an incorrectly answered question x_i, we retain the model's original erroneous response y_i^- and use an external LLM to minimally revise it into a corrected response y_i^+, forming the error-correction set $\mathcal{D}_{\text{err}} = \{(x_i, y_i^-, y_i^+)\}$. During candidate circuit search, the corrected responses in $\mathcal{D}_{\text{err}}$ provide error-correction supervision, while $\mathcal{D}_{\text{clean}}$ is used to preserve the model's existing correct behavior. In practice, we use GSM8K (Cobbe et al., 2021) for mathematical reasoning and MedMCQA (Pal et al., 2022) for medical QA, covering multiple general tasks.

4.2 Error Circuit Discovery

The goal of error circuit discovery is to identify a sparse set of units whose suppression corrects erroneous responses while preserving the model's existing correct behavior. Given the search space $\mathcal{U}$ defined above, RESCUE freezes the original model parameters and assigns each circuit unit $u \in \mathcal{U}$ a trainable mask logit z_u. We use keep-mask semantics, where a mask value of one preserves the unit and a value of zero suppresses its output. Specifically, we define

$$c_u = \text{clip}_{[0,1]}\left(2\sigma\left(\frac{z_0 - z_u}{\tau}\right) - 1\right), \qquad m_u = 1 - c_u \tag{2}$$

where c_u denotes the closure strength and τ is the mask temperature. Initializing $z_u = z_0$ gives $m_u = 1$ for all units, leaving the initial model's computation unchanged. With this initialization, RESCUE first obtains a candidate circuit through reference-based mask optimization, then refines it using generation-level feedback, and finally removes redundant units through circuit pruning.

Candidate Circuit Search. For each error example $(x_i, y_i^-, y_i^+) \in \mathcal{D}_{\text{err}}$, we use the fixed corrected response y_i^+ as the optimization target, while keeping the model parameters frozen. We denote the token-level negative log-likelihood on the corrected responses as $\mathcal{L}_{\text{corr}}$. To preserve the model's existing correct behavior, we also minimize the token-level negative log-likelihood on the clean responses, denoted by $\mathcal{L}_{\text{clean}}$. Let $\mathcal{P}$ denote the set of MLP layer-projection modules, and let $\mathcal{U}_q$ denote the units in module $q \in \mathcal{P}$. The candidate-search objective is

$$\mathcal{L}_{\text{cand}} = \mathcal{L}_{\text{corr}} + \lambda_{\text{clean}}\mathcal{L}_{\text{clean}} + \lambda_{\text{keep}}\frac{1}{|\mathcal{P}|}\sum_{q\in\mathcal{P}}\frac{1}{|\mathcal{U}_q|}\sum_{u\in\mathcal{U}_q} m_u + \mathcal{L}_{\text{sens}}. \tag{3}$$

The third term is a mild keep penalty that encourages mask values to decrease, while the per-module deletion budget introduced below limits the number of closed units. $\mathcal{L}_{\text{sens}}$ favors closing units that improve correction while keeping clean-sensitive units open.

Although continuous keep scores enable gradient-based optimization, they do not directly define the discrete intervention required to isolate an error circuit. Binarizing them only after training could introduce a mismatch between mask optimization and circuit extraction. We therefore employ the *Straight-Through Estimator* (STE) throughout candidate search. At each optimization step, the continuous keep scores are converted into binary keep masks $b_u \in \{0, 1\}$ using a threshold η_0 and a per-module deletion budget ρ_0. Within each module q, at most $\lfloor \rho_0 |\mathcal{U}_q| \rfloor$ units can be closed. If more units satisfy $m_u \le \eta_0$ than allowed by the budget, only those with the smallest keep scores are closed. The binary masks are used in the forward pass, while gradients are propagated through the continuous keep scores using the straight-through estimator:

$$b_u^{\text{ste}} = m_u + \text{sg}(b_u - m_u), \qquad \widetilde{u}(\mathbf{h}) = b_u^{\text{ste}} u(\mathbf{h}). \tag{4}$$

Here, $\text{sg}(\cdot)$ denotes the stop-gradient operator. In the forward pass, $b_u^{\text{ste}} = b_u$, so the model uses binary masks. During backpropagation, gradients flow through m_u, allowing the mask logits to be optimized. Thus, mask optimization uses the same discrete interventions as circuit extraction. After optimization, the units with final binary masks $b_u^{(0)} = 0$ form the candidate circuit $\mathcal{C}_{\text{cand}}$.

RL-Based Circuit Refinement. The reference-based candidate search uses fixed corrected responses and may therefore miss units that sustain errors during free generation. Initialized from the candidate mask $\mathbf{b}^{(0)}$, RESCUE addresses this limitation by sampling multiple responses from the masked model and updating only the mask logits using generation-level outcome feedback.

For each error question x_i with ground-truth answer a_i, the masked model samples a group of K responses $\{\hat{y}_{i,k}\}_{k=1}^K$. We score each response based on final-answer correctness and reasoning

quality, and compute its group-normalized advantage as

$$r_{i,k} = \lambda_{\text{ans}} r_{\text{ans}}(\hat{y}_{i,k}, a_i) + \lambda_{\text{rea}} r_{\text{rea}}(x_i, \hat{y}_{i,k}, a_i), \qquad A_{i,k} = \frac{r_{i,k} - \bar{r}_i}{\max(s_i, \epsilon)}. \tag{5}$$

Here, r_{ans} measures final-answer correctness, r_{rea} is provided by an external reasoning evaluator, and $\bar{r}_i$ and s_i denote the mean and standard deviation of the rewards within the response group. The coefficients λ_{ans} and λ_{rea} balance the two reward components, and ϵ is a small constant for numerical stability. We then optimize the sequence-level objective

$$\mathcal{L}_{\text{RL}} = -\mathbb{E}_i \left[\frac{1}{K} \sum_{k=1}^{K} A_{i,k} \frac{1}{T_{i,k}} \sum_{t=1}^{T_{i,k}} \log p_{\theta, \mathbf{b}^{\text{ste}}} \left(\hat{y}_{i,k,t} \mid x_i, \hat{y}_{i,k,<t} \right) \right]. \tag{6}$$

This group-relative objective increases the likelihood of higher-reward responses and decreases that of lower-reward responses for the same question. It therefore refines the masks using multiple free-running outcomes rather than a single fixed response. During refinement, we retain the STE-based binary masking in Eq. (4) and impose a stage-specific per-module deletion budget. Thus, the model uses discrete masks during generation while gradients are propagated through the continuous masks.

We further stabilize refinement with an auxiliary negative log-likelihood loss on the corrected response y_i^+, assigning it a larger weight when the sampled responses receive uniform rewards and the group-relative signal becomes uninformative. We also compute a negative log-likelihood loss on $\mathcal{D}_{\text{clean}}$, whose gradient helps preserve the model's existing correct behavior. When the refinement gradient conflicts with the clean-preservation gradient, we remove its conflicting component before updating the masks. Mask updates are accepted only when they satisfy both error-correction and clean-performance constraints; otherwise, the model is restored to the last feasible mask. Once the correction target is reached and the clean-performance constraint is satisfied, we progressively tighten the deletion budget to obtain a smaller binary circuit.

Circuit Pruning. The circuit returned by RL may contain unnecessarily closed units, which we prune by reopening them. We rank the closed units by closure strength c_u, from weakest to strongest, and test them for reopening in progressively smaller groups. A group is reopened if the resulting mask preserves error-correction and clean-task performance relative to the RL-refined mask:

$$S_{\text{err}}(\mathbf{b}') \geq S_{\text{err}}(\mathbf{b}^{\text{RL}}) - \epsilon_{\text{err}}, \qquad S_{\text{clean}}(\mathbf{b}') \geq S_{\text{clean}}(\mathbf{b}^{\text{RL}}) - \epsilon_{\text{clean}}. \tag{7}$$

Here, $\mathbf{b}'$ is the proposed mask, and S_{err} and S_{clean} denote the accuracies on error and clean examples, respectively. This process retains only the units whose closure remains necessary. The remaining closed units form the final error circuit $\mathcal{C}^* = \{u \in \mathcal{U} \mid b_u^* = 0\}$.

4.3 Circuit Validation and Tuning

Circuit Ablation Validation. To assess the effect of suppressing the final error circuit $\mathcal{C}^*$, we compare the original model $\mathcal{G}$ with the circuit-ablated model $\mathcal{G} \setminus \mathcal{C}^*$. For $\mathcal{X} \in \{\mathcal{G}, \mathcal{G} \setminus \mathcal{C}^*\}$, we measure normalized final-answer exact-match accuracy on an evaluation set $\mathcal{D}$ as

$$\text{Score}(\mathcal{D}; \mathcal{X}) = \mathbb{E}_{(x,a)\sim\mathcal{D}} \left[\mathbb{I}(\text{Ans}(\mathcal{X}(x)) = a) \right]. \tag{8}$$

A higher score for $\mathcal{G} \setminus \mathcal{C}^*$ on the error set indicates that suppressing the circuit mitigates model failures, while comparable scores on the clean set indicate that existing correct behavior is largely preserved. Together, these scores assess repair effectiveness and behavioral preservation.

Circuit-Restricted Tuning. Given the final error circuit $\mathcal{C}^*$, we restrict model updates to its corresponding parameters. For each unit $u = (l, p, j) \in \mathcal{C}^*$, where $p \in \{\text{gate}, \text{up}, \text{down}\}$, we tune the j-th output row of projection p in layer l, together with its bias if present. For each selected row, we introduce a zero-initialized trainable weight delta, together with a bias delta if present, while freezing all original model parameters. We optimize the selected deltas on the error-correction set using the corrected responses to supervise both intermediate reasoning and final-answer prediction:

$$\min_{\Delta\theta(\mathcal{C}^*)} -\mathbb{E}_{(x,y^+)\sim\mathcal{D}_{\text{err}}} \left[\frac{\sum_t w_t \log p_{\theta+\Delta\theta(\mathcal{C}^*)} \left(y_t^+ \mid x, y_{<t}^+ \right)}{\sum_t w_t} \right]. \tag{9}$$

where w_t assigns greater weight to final-answer tokens. After training, we merge the best-performing deltas into the selected projection parameters, leaving all other parameters unchanged.

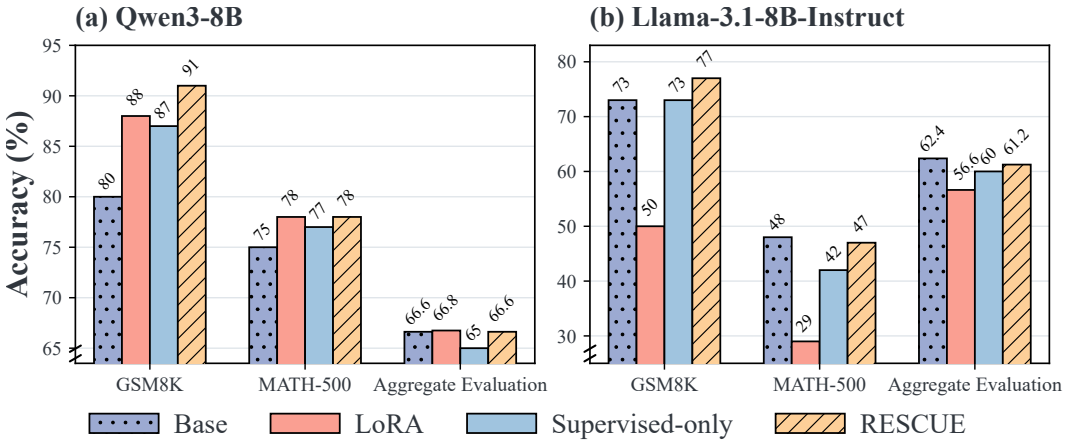


Figure 2: Post-tuning accuracy after heterogeneous GSM8K repair; Aggregate Evaluation is the macro-average over the eight non-target benchmarks in Section 5.1. The panels use different truncated y-axis ranges for readability, with exact percentages shown above each bar.

5 Experiments

5.1 Experimental Settings

Models. We evaluate RESCUE on two instruction-tuned language models with different architectures: Qwen3-8B (Yang et al., 2025) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024).

Datasets. Our main experiments use GSM8K (Cobbe et al., 2021). We construct five *pattern-specific* repair datasets, each with a $4:1$ train–test split. Examples within each dataset share an underlying reasoning structure while varying in surface entities and numerical values. We additionally construct a *heterogeneous* GSM8K repair dataset with a $3:1$ train–test split. For robustness evaluation, we create rephrased versions of five pattern-specific datasets, retaining the same split ratio. We further evaluate cross-domain repair on MedMCQA (Pal et al., 2022) with the same setting.

Baselines and Controls. We compare RESCUE with the base model, direct LoRA fine-tuning (Hu et al., 2022), and a *supervised-only* variant that uses the circuit obtained before RL refinement and pruning. All localized circuits undergo the same ablation and circuit-restricted tuning procedures. We additionally evaluate ten density-matched random circuits as controls.

Evaluation. We evaluate each localized circuit through ablation and circuit-restricted tuning. For mathematical reasoning, we evaluate target-task performance on GSM8K (Cobbe et al., 2021) and generalization on MATH-500 (Hendrycks et al., 2021b; Lightman et al., 2024). We assess capability preservation on ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), PIQA (Bisk et al., 2020), TruthfulQA-MC1 (Lin et al., 2022), the High School Mathematics and High School Statistics subsets of MMLU (Hendrycks et al., 2021a), and the Multi-Step Arithmetic task from BBH (Suzgun et al., 2023). For cross-domain repair, we use MedMCQA (Pal et al., 2022) as the target and evaluate capability retention on the same eight benchmarks. All public benchmark evaluations are conducted with the Language Model Evaluation Harness (Gao et al., 2024), using identical configurations across models to enable comparisons.

Metrics. After circuit ablation, we report full-set repair accuracy on the constructed datasets and GSM8K accuracy. After circuit-restricted tuning, we report benchmark accuracy and its absolute change from the base model. Circuit density is measured as the percentage of selected MLP projection rows among all eligible rows. Random-circuit ablation results are aggregated over ten density-matched samples. Further evaluation details are provided in Appendix A.

5.2 Heterogeneous Repair

We evaluate whether RESCUE can repair a heterogeneous collection of GSM8K reasoning failures rather than one error pattern. On the mixed repair set, we compare RESCUE with the base model, di-

Table 1: Pattern-specific circuit ablation. Circuit denotes the proportion of selected MLP rows.

Model	Error Pattern	Base (Acc.)	RESCUE Abl. (Acc.)	Δ (pp)	Circuit
Qwen3-8B	Misleading Premise Back-solving	2%	**93%**	**+91**	0.32%
	Dynamic State Tracking	1%	**91%**	**+90**	0.31%
	Dropped-lowest Average	4%	**94%**	**+90**	0.26%
	Declining Group Accumulation	2%	**76%**	**+74**	0.44%
	Multiplicative Increment Back-solving	0%	**91%**	**+91**	0.39%
Llama-3.1-8B-Instruct	Cumulative Target Gap	1%	**95%**	**+94**	0.62%
	Scale Conversion	36%	**92%**	**+56**	0.74%
	Percentage Risk	6%	**95%**	**+89**	0.60%
	Unit Price and Total Cost	31%	**95%**	**+64**	0.63%
	Fees and Change	24%	**94%**	**+70**	0.60%

rect LoRA, and the supervised baseline. Figure 2 reports post-tuning accuracy on GSM8K, MATH-500, and Aggregate Evaluation, defined as the macro-average over the eight non-target benchmarks.

Target-task improvement. Before tuning, circuit ablation raises full-set repair accuracy from 0% to 50.5% on Qwen3-8B and from 6.0% to 69.0% on Llama-3.1-8B-Instruct, using only 1.49% and 1.40% of the eligible MLP rows, respectively. Restricting tuning to their parameters then yields the highest GSM8K accuracy on both models. On Qwen3-8B, it improves accuracy from 80% to 91%, outperforming LoRA (88%) and supervised-only (87%) by 3 and 4 percentage points, respectively. On Llama-3.1-8B-Instruct, RESCUE improves accuracy from 73% to 77%, whereas supervised-only remains at 73% and LoRA decreases to 50%. The advantage over supervised-only indicates that the RESCUE pipeline—output-guided mask refinement followed by pruning—produces circuits that are more effective for downstream repair. Circuit-restricted tuning also raises full-set repair accuracy to 53.5% and 75.5% on the two models, respectively.

Generalization and capability preservation. RESCUE largely preserves mathematical generalization and non-target capabilities. On Qwen3-8B, it improves MATH-500 from 75% to 78% and retains Aggregate Evaluation at 66.6%. On Llama-3.1-8B-Instruct, it retains 47% on MATH-500 and 61.2% on Aggregate Evaluation, versus base scores of 48% and 62.4%, while supervised-only falls to 42% and 60.0% and LoRA to 29% and 56.6%. Detailed results are provided in Appendix B.1. Overall, RESCUE has a trade-off between repairing failures and preserving capabilities.

5.3 Pattern-Specific Error Circuits and Attribution Specificity

Pattern-specific localization and repair. To test whether RESCUE localizes components that causally contribute to recurring failures, we independently identify and ablate circuits for five pattern-specific GSM8K repair sets per model. Table 1 reports full-set accuracy before and after ablation, together with the proportion of selected MLP rows. Across the five independently tuned circuits, Aggregate Evaluation differs from the base model by at most 0.75 percentage points for Qwen3-8B and 1.25 points for Llama-3.1-8B-Instruct, indicating that pattern-specific repair largely preserves broader capabilities. Detailed results are provided in Appendix B.2.

Ablating the localized circuits consistently corrects the corresponding failures using highly sparse interventions. On Qwen3-8B, accuracy increases from 0–4% to 76–94%, corresponding to gains of 74–91 percentage points with circuits comprising only 0.26–0.44% of the candidate rows. On Llama-3.1-8B-Instruct, accuracy increases from 1–36% to 92–95%, with gains of 56–94 points using only 0.60–0.74% of the candidate rows. These results provide causal evidence that sparse internal components contribute to recurring reasoning failures.

Density-matched attribution controls. The gains above might still arise from applying the repair procedure to any sparse subset of the same size, which would make circuit attribution unnecessary. To test this possibility, we initialize ten per-tensor density-matched random circuit masks for two patterns per model and apply the same supervised mask-optimization and RL-refinement stages. We then ablate the resulting circuits and evaluate them on the corresponding examples subsets. All scores are ablation results; no circuit-restricted tuning is performed.

Despite selecting the same number of rows and using the same mask-optimization procedure, the random-circuit controls remain close to the base model across all four settings, with mean accu-

Table 2: Cross-rephrase transfer of pattern-specific error circuits. Accuracy is measured over 100 rephrased examples before and after ablating the circuit localized from the original formulation.

Model	Error Pattern	Rephrased(Acc.)	Ablation (Acc.)	Δ (pp)
Qwen3-8B	Misleading Premise Back-solving	28%	**42%**	**+14**
	Dynamic State Tracking	17%	**25%**	**+8**
	Dropped-lowest Average	3%	**16%**	**+13**
	Declining Group Accumulation	1%	**18%**	**+17**
	Multiplicative Increment Back-solving	0%	**23%**	**+23**
Llama-3.1-8B-Instruct	Cumulative Target Gap	25%	**60%**	**+35**
	Scale Conversion	35%	**53%**	**+18**
	Percentage Risk	26%	**92%**	**+66**
	Unit Price and Total Cost	42%	**94%**	**+52**
	Fees and Change	15%	**75%**	**+60**

racy ranging from 0.0% to 13.5% and a maximum individual result of 30%. In contrast, RESCUE achieves 90–95% on the same subsets. For Llama-3.1-8B-Instruct on Pattern 1, all ten random-circuit controls obtain 0%, whereas RESCUE reaches 95%. These results show that neither circuit density nor the repair procedure alone is sufficient; effective repair depends on first attributing the failure to functionally relevant model components rather than selecting arbitrary sparse subsets.

5.4 Cross-Rephrase Transfer of Error Circuits

Cross-rephrase evaluation. Pattern-specific datasets share similar surface forms, raising the concern that the localized circuits may capture template-specific lexical patterns rather than the underlying failure-producing computation. We therefore rephrase each dataset while preserving its reasoning structure and directly ablate, on the rephrased inputs, the circuit localized from the original formulation. No circuit re-localization, mask refinement, or parameter tuning is performed. Table 2 reports accuracy over all 100 rephrased examples before and after this same-circuit ablation.

Results. Same-circuit ablation improves accuracy for all ten model–pattern combinations. The gains range from 8 to 23 percentage points for Qwen3-8B, with an average of 15.0 points, and from 18 to 66 points for Llama-3.1-8B-Instruct, with an average of 46.2 points. Since the circuits are localized only from the original formulations, these consistent gains indicate that their causal effects transfer across surface reformulations. This suggests RESCUE captures relevant components linked with recurring reasoning failures rather than template-specific lexical mappings, although varying gains suggest that the same circuit need not account equally for every linguistic realization. Complementary results obtained by re-localizing circuits on the rephrased datasets appear in Appendix B.3.

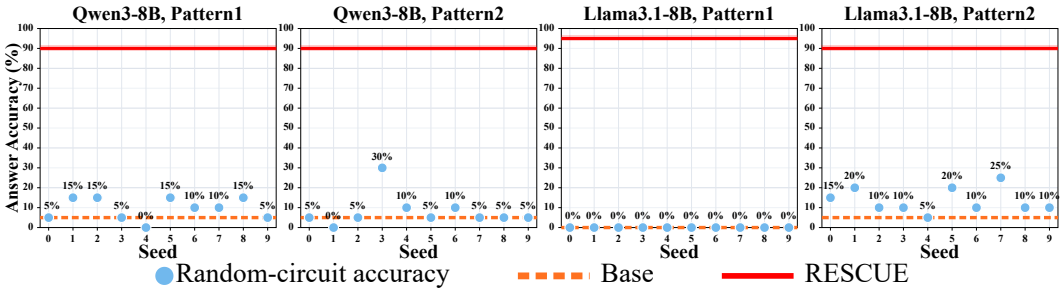


Figure 3: Density-matched random-circuit controls. Pale-blue lines denote their means, while orange dashed and red solid lines denote Base and RESCUE, respectively.

5.5 Sequential Repair Preserves Earlier Repairs

Sequential repair. In practical deployment, model failures may be discovered incrementally rather than collected in a single repair set. We therefore apply RESCUE sequentially to five pattern-specific GSM8K datasets. Starting from base model M_0, at stage t we repair the next error pattern

Table 3: Cross-domain repair results on MedMCQA. Repair-set accuracy is measured after RESCUE and after LoRA. Aggregate Eval. is the mean accuracy over the eight benchmarks.

		Targeted Repair		Post-tuning Evaluation	
Model	**Method**	**Repair-set (Acc.)**	**Circuit**	**MedMCQA**	**Aggregate Eval.**
Qwen3-8B	Base	3.0%	–	59%	66.63%
	LoRA	65.0%	–	**60%**	64.75%
	RESCUE	**78.5%**	**1.47%**	**60%**	**66.75%**
Llama-3.1-8B-Instruct	Base	0.0%	–	**59%**	**62.38%**
	LoRA	65.5%	–	57%	58.88%
	RESCUE	**81.0%**	**1.44%**	**59%**	59.75%

using M_{t-1}, yielding M_t. After each stage, we evaluate M_t on all five pattern sets; Figure 4 reports full-set ablation accuracy across repair stages and error patterns.

Results. Mean accuracy across the five pattern sets increases steadily from 1.8% to 90.8% for Qwen3-8B and from 19.6% to 92.6% for Llama-3.1-8B-Instruct. At their respective repair stages, the targeted patterns reach 93%, 100%, 97%, 94%, and 76% for Qwen3-8B, and 95%, 87%, 100%, 99%, and 100% for Llama-3.1-8B-Instruct. Earlier repairs are largely retained: after all five stages, the corresponding accuracies remain 99%, 99%, 89%, 91%, and 76% for Qwen3-8B, and 91%, 79%, 94%, 99%, and 100% for Llama-3.1-8B-Instruct, with a maximum later-stage decrease of only 8 percentage points for either model. These results show that RESCUE can accumulate repairs for newly discovered failures while introducing only limited interference with previous corrections. Post-tuning benchmark results after the final repair stage are provided in Appendix B.4.

5.6 Cross-Domain Repair on MedMCQA

Cross-domain evaluation. To evaluate cross-domain repair, we apply RESCUE and direct LoRA to heterogeneous MedMCQA repair sets for both models. Table 3 reports full-set repair accuracy, circuit size, and benchmark performance on MedMCQA and Aggregate Evaluation.

Results. Across the two models, RESCUE raises repair-set accuracy from 0–3% to 78.5–81.0%, outperforming LoRA (65.0–65.5%) while selecting only 1.44–1.47% of the candidate MLP rows. After tuning, RESCUE maintains or improves MedMCQA accuracy and better preserves aggregate performance than LoRA. These results show that RESCUE extends beyond mathematical reasoning, enabling sparse targeted repair while better preserving broader capabilities than direct fine-tuning. Full per-benchmark results are provided in Appendix B.5.

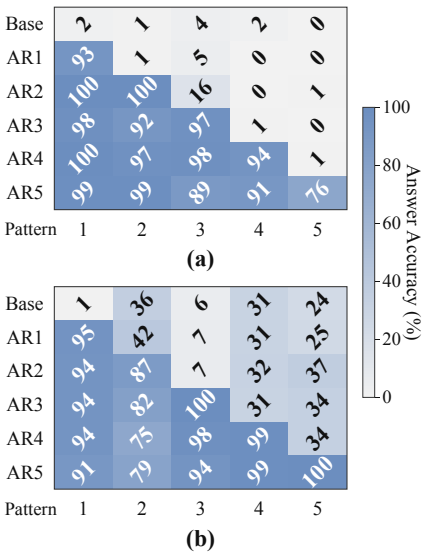


Figure 4: Full-set ablation accuracy (%) during sequential repair for (a) Qwen3-8B and (b) Llama-3.1-8B-Instruct. AR1–AR5 denote repair stages; pattern order follows Table 1.

6 Conclusion

In this work, we introduce **RESCUE**, a framework for localizing sparse circuits associated with model errors and restricting repair to their parameters. RESCUE combines reference-based mask optimization with outcome-guided refinement over multiple model generations, followed by circuit pruning and targeted tuning. Experiments on Qwen3-8B and Llama-3.1-8B-Instruct across mathematical and medical tasks show that recurring errors are associated with highly sparse MLP circuits whose ablation improves task performance. RESCUE further outperforms the supervised-only baseline and LoRA while largely preserving non-target capabilities. Specifically, on heterogeneous GSM8K repair, RESCUE improves accuracy by 4 percentage points over supervised-only localization on both models and by up to 27 points over LoRA. Our findings extend circuit-based mechanistic interpretability beyond behavior analysis and safety interventions to targeted repair.

AI USE STATEMENT

Generative AI tools were used to produce the `corrected_reasoning` field in our dataset, assist with literature retrieval, review manuscript formatting, and suggest caption and editorial revisions. All AI-assisted outputs and suggestions were reviewed by the authors. The authors take full responsibility for the final content of this work, including all text, data, results, and claims.

REPRODUCIBILITY STATEMENT

To support reproducibility, we release our implementation in an anonymous repository. The repository includes code for supervised mask optimization, reinforcement-learning-based mask refinement, circuit pruning, circuit ablation, circuit-restricted tuning, and benchmark evaluation. Detailed data-construction procedures, prompts, experimental protocols, and evaluation metrics are provided in the Appendix. Together, these resources support reproduction of the main experiments and analyses reported in this paper.

REFERENCES

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.68. URL https://aclanthology.org/2023.findings-emnlp.68/.

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. Curran Associates, Inc., 2024. doi: 10.52202/079017-0587. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/20fdaf67581e6d7157376d1ed584040a-Abstract-Conference.html.

Tian Bian, Yifan Niu, Chaohao Yuan, Chengzhi Piao, Bingzhe Wu, Long-Kai Huang, Yu Rong, Tingyang Xu, Hong Cheng, and Jia Li. IBCircuit: Towards holistic circuit discovery with information bottleneck. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4289–4302. PMLR, 2025. URL https://proceedings.mlr.press/v267/bian25a.html.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://doi.org/10.1609/aaai.v34i05.6239.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda, 2023. OpenReview.net. URL https://openreview.net/forum?id=ETKGuby0hcs.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, Vienna, Austria, 2024. OpenReview.net. URL https://openreview.net/forum?id=Zj12nzlQbz.

Hakaze Cho, Mariko Kato, Yoshihiro Sakai, and Naoya Inoue. Revisiting in-context learning inference circuit in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xizpnYNvQq.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. DoLa: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*, Vienna, Austria, 2024. OpenReview.net. URL https://openreview.net/forum?id=Th6NyL07na.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Abstract-Conference.html.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 8493–8502. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.581. URL https://aclanthology.org/2022.acl-long.581/.

Cheng Gao, Huimin Chen, Chaojun Xiao, Zhiyi Chen, Zhiyuan Liu, and Maosong Sun. H-neurons: On the existence, impact, and origin of hallucination-associated neurons in llms, 2025. URL https://arxiv.org/abs/2512.01797.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alexis Le Noac'h, Haonan Li, Kieran McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, July 2024. URL https://zenodo.org/records/12608602. Software, version 0.4.3.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 30–45. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.3. URL https://aclanthology.org/2022.emnlp-main.3/.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969*, 2023. doi: 10.48550/arXiv.2304.05969. URL https://arxiv.org/abs/2304.05969.

Aaron Grattafiori et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Michael Hanna, Ollie Liu, and Alexandre Variengien. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 76033–76060. Neural Information Processing Systems Foundation, 2023. doi: 10.52202/075280-3322. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/efbba7719cc5172d175240f24be11280-Abstract-Conference.html.

Michael Hanna, Mateusz Piotrowski, Jack Lindsey, and Emmanuel Ameisen. Circuit-tracer: A new library for finding feature circuits. In Yonatan Belinkov, Aaron Mueller, Najoung Kim, Hosein Mohebbi, Hanjie Chen, Dana Arad, and Gabriele Sarti (eds.), *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 239–249, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-346-3. doi: 10.18653/v1/2025.blackboxnlp-1.14. URL https://aclanthology.org/2025.blackboxnlp-1.14/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021b. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Guan Zhe Hong, Nishanth Dikkala, Enming Luo, Cyrus Rashtchian, Xin Wang, and Rina Panigrahy. A implies b: Circuit analysis in LLMs for propositional logical reasoning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 30749–30803. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/2c4477572f06102b2ae3bd2b2ffcebf9-Abstract-Conference.html.

Yifan Hou, Jiaoda Li, Yu Fei, Alessandro Stolfo, Wangchunshu Zhou, Guangtao Zeng, Antoine Bosselut, and Mrinmaya Sachan. Towards a mechanistic interpretation of multi-step reasoning capabilities of language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4902–4919. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.299. URL https://aclanthology.org/2023.emnlp-main.299/.

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. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 15696–15707. PMLR, 2023. URL https://proceedings.mlr.press/v202/kandpal23a.html.

Dmitrii Kharlapenko, Stepan Shabalin, Arthur Conmy, and Neel Nanda. Scaling sparse feature circuits for studying in-context learning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Mp8Og8Rpop.

Kenneth Li, Oam Patel, Fernanda B. Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 41451–41530. Neural Information Processing Systems Foundation, 2023. doi: 10.52202/075280-1797. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/81b8390039b7302c909cb769f8b6cd93-Abstract-Conference.html. Spotlight presentation.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, 2022. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=I4e82CIDxv.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp.

17359–17372, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Yaniv Nikankin, Anja Reusch, Aaron Mueller, and Yonatan Belinkov. Arithmetic without algorithms: Language models solve math with a bag of heuristics. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=O9YTt26r2P.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *The Thirteenth International Conference on Learning Representations*, Singapore, 2025. OpenReview.net. URL https://openreview.net/forum?id=KRnsX5Em3W.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260. PMLR, 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, August 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

Yucheng Sun, Alessandro Stolfo, and Mrinmaya Sachan. Probing for arithmetic errors in language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8111–8128, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.411. URL https://aclanthology.org/2025.emnlp-main.411/.

Zhongxiang Sun, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Weijie Yu, Yang Song, and Han Li. ReDeEP: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability. In *The Thirteenth International Conference on Learning Representations*, Singapore, 2025b. OpenReview.net. URL https://openreview.net/forum?id=ztzZDzgfrh.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 407–416, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.25. URL https://aclanthology.org/2024.blackboxnlp-1.25/.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. Knowledge circuits in pretrained transformers. In *Advances in Neural Information Processing Systems*, volume 37, pp. 118571–118602. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d6df31b1be98e04be48af8bedb95b499-Abstract-Conference.html.

Lei Yu, Meng Cao, Jackie Chi Kit Cheung, and Yue Dong. Mechanistic understanding and mitigation of language model non-factual hallucinations. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7943–7956, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.466. URL https://aclanthology.org/2024.findings-emnlp.466/.

Miao Yu, Fanci Meng, Xinyun Zhou, Shilong Wang, Junyuan Mao, Linsey Pan, Tianlong Chen, Kun Wang, Xinfeng Li, Yongfeng Zhang, et al. A survey on trustworthy llm agents: Threats and countermeasures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 6216–6226, 2025a.

Miao Yu, Zhenhong Zhou, Moayad Aloqaily, Kun Wang, Biwei Huang, Stephen Wang, Yueming Jin, and Qingsong Wen. Backdoor attribution: Elucidating and controlling backdoor in language models. *arXiv preprint arXiv:2509.21761*, 2025b.

Miao Yu, Siyuan Fu, Moayad Aloqaily, Zhenhong Zhou, Safa Otoum, Xing fan, Kun Wang, Yufei Guo, and Qingsong Wen. Safeseek: Universal attribution of safety circuits in language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=XhkRBQd8gw.

Hongbang Yuan, Pengfei Cao, Zhuoran Jin, Yubo Chen, Daojian Zeng, Kang Liu, and Jun Zhao. Whispers that shake foundations: Analyzing and mitigating false premise hallucinations in large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2670–2683, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.155. URL https://aclanthology.org/2024.emnlp-main.155/.

Mert Yuksekgonul, Varun Chandrasekaran, Erik Jones, Suriya Gunasekar, Ranjita Naik, Hamid Palangi, Ece Kamar, and Besmira Nushi. Attention satisfies: A constraint-satisfaction lens on factual errors of language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gfFVATffPd.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Shaolei Zhang, Tian Yu, and Yang Feng. TruthX: Alleviating hallucinations by editing large language models in truthful space. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8908–8949, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.483. URL https://aclanthology.org/2024.acl-long.483/.

Wei Zhang, Chaoqun Wan, Yonggang Zhang, Yiu ming Cheung, Xinmei Tian, Xu Shen, and Jieping Ye. Interpreting and improving large language models in arithmetic calculation. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=CfOtiepP8s.

Yiran Zhao, Wenxuan Zhang, Yuxi Xie, Anirudh Goyal, Kenji Kawaguchi, and Michael Shieh. Understanding and enhancing safety mechanisms of LLMs via safety-specific neuron. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yR47RmND1m.

A Experimental Details

A.1 Dataset Construction

Our experiments use paired error-correction and clean sets with complementary roles. For each model, we first run the unmodified model on candidate questions under the task-specific prompt in Appendix A.2. Incorrect generations provide the source examples for repair, whereas correctly answered questions provide clean examples. For each incorrect generation, we retain the question, ground-truth answer, original model response, parsed prediction, and correctness label. Gemini 3.1 Pro then produces a minimally corrected reasoning trace that preserves valid intermediate steps and changes only the content needed to recover the ground-truth answer. The resulting triples of question, erroneous response, and corrected response form the error-correction set used for circuit localization and tuning. All localization and tuning data originate from the official training splits and are disjoint from benchmark evaluation data.

Pattern-specific GSM8K sets. For each model, we construct five datasets, each centered on one recurring reasoning failure. The Qwen3-8B patterns are *Misleading Premise Back-solving*, *Dynamic State Tracking*, *Dropped-lowest Average*, *Declining Group Accumulation*, and *Multiplicative Increment Back-solving*. The Llama-3.1-8B-Instruct patterns are *Cumulative Target Gap*, *Scale Conversion*, *Percentage Risk*, *Unit Price and Total Cost*, and *Fees and Change*.

Starting from a representative failure, Gemini 3.1 Pro generates variants that preserve the underlying sequence of mathematical operations while varying names, locations, numerical values, background scenarios, and wording. Gemini is also used to verify the generated answer and reasoning trace and to identify duplicate, invalid, or structurally inconsistent items. Samples that do not preserve the intended solution logic are excluded. Each finalized dataset contains 100 examples.

Rephrased sets. For cross-rephrase transfer, Gemini 3.1 Pro rewrites the pattern-specific questions while preserving their entities, numerical conditions, mathematical relationships, solution method, and final answer. The rewrite changes sentence structure, voice, modifier placement, and narrative organization. Each rephrased dataset contains 100 examples corresponding one-to-one with its original-formulation dataset. The main-text transfer experiment applies the circuit localized on the original formulation directly to these rephrased inputs, without re-localization or parameter tuning. Appendix B.3 additionally reruns the complete localization procedure on two rephrased patterns per model.

Sequential and cross-domain sets. Sequential repair uses the same five pattern-specific datasets in the within-model order reported in the main paper. At each stage, the next pattern is repaired from the model state produced by the preceding stage, after which all five pattern sets are reevaluated. For cross-domain evaluation, we construct a MedMCQA repair set separately for each model. MedMCQA examples retain the multiple-choice options and ground-truth option, while their corrected references explain the selected medical answer under the same minimum-intervention principle.

A.2 Prompt Templates

We use Gemini 3.1 Pro for three data-construction operations: generating variants of a recurring problem type, rephrasing existing questions without changing their semantics, and minimally correcting erroneous model reasoning. The following boxes provide faithful English translations of the original Chinese prompts while preserving their instructions and constraints. Answer verification and deduplication are conducted as follow-up checks within the same interaction rather than with separately fixed templates. Placeholders such as `[EXAMPLE]`, `[PROBLEMS]`, and `[SAMPLES]` are replaced at runtime with the corresponding source items.

Illustrative example. Figure 8 illustrates the minimum-intervention correction process. Only the erroneous calculation and its downstream result are revised, while the remaining reasoning is preserved.

Task and reward prompts. The following boxes report the task-generation prompts and reasoning-evaluation prompts used during RL refinement. The task prompts are provided as sys-

Pattern-Specific Problem Generation Prompt

```
Given the following example problem, generate 100 similar problems. Change incidental details
such as names, locations, and numbers, and introduce some diversity, but do not change the
overall solution logic.
While preserving the original solution procedure, diversify the background, names, scenarios,
numerical values, answers, and wording.
Return the results in JSON format and assign each item an ID from 1 to 100.

Example:
[EXAMPLE]
```

Figure 5: English translation of the prompt for generating pattern-specific GSM8K problems.

Problem Rephrasing Prompt

```
I will provide a set of problem samples.
Rewrite only their wording and presentation without changing their meaning, logic, or solution
method.
Thoroughly change the sentence structure, voice, modifier placement, and narrative organization
so that each item reads like a newly written problem while exactly preserving
the original mathematical logic.
Do not freely alter any problem. Preserve the names, locations, numerical values, mathematical
relationships,  solution logic, and answer. The rewritten problem must be fully equivalent to
the original, rather than merely yielding the same final answer.

Problems:
[PROBLEMS]
```

Figure 6: English translation of the prompt for rephrasing pattern-specific problems.

tem messages to the repaired model, whereas the evaluation prompts are provided to Qwen2.5-14B-Instruct, which assigns the reasoning reward. The same prompt is used for Qwen3-8B and Llama-3.1-8B-Instruct within each domain. The same output contracts are also retained across RESCUE and the corresponding comparison methods.

GSM8K Generation System Prompt

```
Solve the following grade-school math problem.
Use concise reasoning. Do not write an introduction.
Do not use Markdown headings.
Track the changing quantities carefully.
Your response must follow this exact format:

Reasoning:
1. <short calculation>
2. <short calculation>
3. <short calculation>
4. <short calculation>
Final Answer: <only one number>

Stop immediately after the Final Answer line.
```

Minimal-Intervention Correction Prompt

```
I will next provide samples containing erroneous model outputs. Correct their erroneous
outputs according to the minimum-intervention principle.
Preserve every valid reasoning step and modify only the content necessary to correct the error.
Place the minimally modified response in the "corrected_reasoning" field. Return only the
following fields in JSON format:

"id"
"question"
"ground_truth"
"generation"
"pred_answer_masked"
"is_correct_masked"
"corrected_reasoning"

Samples:
[SAMPLES]
```

Figure 7: English translation of the prompt for minimally correcting erroneous reasoning.

Illustrative Minimal-Intervention Correction

Question. Parker buys a $100 string trimmer and a leaf blower costing twice as much. Both items receive a 20% discount, after which she buys $50 safety ear muffs. How much does she spend in total?

Original erroneous output.

> ...Since Parker gets a 20% discount, the leaf blower costs $200-$20=$180, while the string trimmer costs $100-$20=$80. Therefore, Parker spends $180+$80+$50=$310.

Minimally corrected output.

> ...Since Parker gets a 20% discount, the leaf blower costs $200-$40=$160, while the string trimmer costs $100-$20=$80. Therefore, Parker spends $160+$80+$50=$290.

Ground-truth answer: $290.

Figure 8: Illustrative minimum-intervention correction. Colored text marks the erroneous and revised quantities; unchanged reasoning is omitted for brevity.

GSM8K Reasoning-Evaluation Prompt

```
You are a strict math-solution judge.
Evaluate the model response with one score chosen ONLY from {0, 0.25, 0.5, 0.75, 1.0}.
Return only one JSON object in this exact schema:
{"reasoning_score": 0.75}

Scoring rules:
- reasoning_score: 1.0 means fully correct reasoning and a consistent final answer;
0 means wrong or nonsensical reasoning; intermediate values indicate
  partial correctness.
- Grade the visible reasoning and its consistency with the final answer.

Question:
[QUESTION]

Ground truth answer:
[GROUND_TRUTH]

Model response:
[MODEL_RESPONSE]
```

MedMCQA Generation System Prompt

```
Answer the medical multiple-choice question.
Use concise reasoning. Do not write an introduction.
Do not use Markdown headings.
End with exactly one final answer line in the requested format.
Stop immediately after the Final Answer line.
```

MedMCQA Reasoning-Evaluation Prompt

```
You are a strict medical multiple-choice reasoning judge.
Evaluate the model response with one score chosen ONLY from {0, 0.25, 0.5, 0.75, 1.0}.
Return only one JSON object in this exact schema:
{"reasoning_score": 0.75}

Scoring rules:
- reasoning_score: 1.0 means the reasoning supports the correct medical option and is
consistent with the final answer.
- reasoning_score: 0 means wrong, unsafe, nonsensical, or inconsistent with the
correct option.
- Intermediate values indicate partially correct reasoning.

Question:
[QUESTION]

Ground truth option:
[GROUND_TRUTH]

Model response:
[MODEL_RESPONSE]
```

Prompt instantiation and reward parsing. For the GSM8K reasoning judge, `[QUESTION]`, `[GROUND_TRUTH]`, and `[MODEL_RESPONSE]` are replaced by the current problem, its numerical answer, and one sampled masked-model response. The MedMCQA judge receives the corresponding multiple-choice question, ground-truth option, and sampled response. Both judges are instructed to return a JSON object containing only `reasoning_score`. The parsed score is discretized to one of $\{0, 0.25, 0.5, 0.75, 1.0\}$ and combined with the independently computed final-answer reward in the group-relative objective of Section 4.2.

GSM8K responses terminate with a single numerical answer, whereas MedMCQA responses terminate with the selected option label and option text. Final-answer correctness is computed independently of the reasoning judge by extracting the final-answer field and comparing it with the normalized ground truth. Separating answer correctness from reasoning quality allows partially correct reasoning to provide a graded learning signal when the final answer is wrong.

A.3 Implementation Details

Mask parameterization. We apply masks only to output rows of the MLP gate, up, and down projections and leave all attention modules unchanged. Each eligible row has one trainable mask logit, initialized to 0.2. The original language-model parameters remain frozen throughout circuit localization. Following Section 4.2, the forward pass uses binary keep masks, while the straight-through estimator propagates gradients through the underlying continuous mask values.

Supervised candidate localization. The supervised stage is run for 10 epochs. Corrected responses from the error-correction set provide the correction objective, while original responses from the clean set provide the preservation objective. We evaluate the binary mask during localization and retain the best feasible checkpoint rather than relying only on the final continuous mask state. At the end of this stage, all rows assigned a zero keep mask constitute the supervised candidate circuit. The supervised-only baseline uses this pre-RL, pre-pruning circuit and subsequently undergoes the same ablation and circuit-restricted tuning procedures as RESCUE.

Outcome-guided refinement. RL refinement starts from the supervised candidate mask and is run for 6 epochs. For every repair example, we sample four masked-model responses and score each response using final-answer correctness and the graded reasoning score described above. Rewards are normalized within the responses generated for the same question. The refinement stage also retains an auxiliary likelihood objective on the minimally corrected reference and a clean-set preservation objective. When the correction and clean-preservation gradients conflict, the implementation projects away the conflicting component before updating the mask. Candidate updates are accepted only when they satisfy the correction and clean-performance constraints, and the last feasible binary mask is restored when a proposed update violates them.

Pruning and circuit-restricted tuning. After RL refinement, closed rows are ranked by closure strength and progressively considered for reopening. A group is reopened only when both repair and clean-set performance remain within the specified tolerances of the refined mask. This procedure removes rows whose suppression is no longer necessary and yields the final circuit used in the reported ablation experiments. Circuit-restricted tuning then introduces zero-initialized trainable deltas only for the selected gate, up, and down projection rows. All original parameters and all rows outside the circuit remain frozen. Training uses the minimally corrected responses, with additional weight on final-answer tokens; the best-performing deltas are finally merged into the selected rows.

Baselines and random controls. Direct LoRA applies conventional low-rank fine-tuning to the same repair data without circuit localization. The supervised-only baseline omits both RL mask refinement and circuit pruning but uses the same downstream validation and restricted-tuning pipeline. For the density-matched control, we initialize ten random per-tensor masks with the same number of selected rows as the attributed circuit and apply the same supervised mask optimization and RL refinement. We then perform circuit ablation and evaluate each control on the corresponding subset.

Precision and reproducibility. All experiments use BF16 precision and a fixed random seed of 42, except that the density-matched analysis explicitly evaluates ten random mask seeds. For Qwen3-8B, the model-specific extended thinking mode is disabled so that both model families follow the visible reasoning format given in Appendix A.2. Identical task prompts and benchmark configurations are used across methods within each model. Complete optimizer settings, checkpoint-selection thresholds, pruning tolerances, and baseline configurations are included in the accompanying code.

A.4 Evaluation and Reporting Conventions

Public benchmarks. For GSM8K repair, we first evaluate the circuit-ablated model on GSM8K before any parameter update. After circuit-restricted tuning, we evaluate GSM8K, MATH-500, and the eight non-target benchmarks listed in Section 5.1 with the Language Model Evaluation Harness under identical task configurations. For MedMCQA repair, MedMCQA is the target public benchmark and GSM8K is reported separately as an additional cross-domain retention measure. *Aggregate Evaluation* is the unweighted macro-average of ARC-Challenge, HellaSwag, WinoGrande, PIQA, TruthfulQA-MC1, MMLU High School Mathematics, MMLU High School Statistics, and BBH Multi-Step Arithmetic. GSM8K, MATH-500, and MedMCQA are not included.

For constructed GSM8K repair sets, numerical answers are extracted from the designated final-answer field and compared with normalized ground-truth answers. For MedMCQA, correctness is determined from the selected option. Public-benchmark scores follow the corresponding Language Model Evaluation Harness task implementation.

Circuit density. Circuit density is the number of selected MLP projection rows divided by the total number of eligible gate, up, and down projection rows. The denominator is computed separately for each model architecture. All percentages therefore describe a fraction of the candidate MLP-row search space.

Specialized controls. The cross-rephrase transfer experiment evaluates all 100 rephrased examples using the original-formulation circuit without re-localization, mask refinement, or tuning. The re-localization experiment in Appendix B.3 instead reruns the complete RESCUE pipeline on the rephrased data before ablation and tuning. In sequential repair, AR1–AR5 denote the model states

obtained after stages 1–5, and every state is evaluated on all five pattern sets to expose both accumulation and forgetting. In the random-circuit control, each point is an ablation accuracy from one density-matched seed; the mean summarizes the ten seeds, while the spread reflects sensitivity to random circuit initialization.

B Additional Experimental Results

This section provides the complete benchmark-level results that underlie the aggregate quantities reported in the main paper. Repair-set accuracy measures whether the constructed failures are corrected, whereas the public benchmarks measure the behavior of the post-tuning model outside those examples. Aggregate Evaluation is the unweighted mean over the eight non-target benchmarks defined in Appendix A.4.

Aggregate capability retention. To summarize capability preservation across experiments, we compute the retention ratio between the post-tuning Aggregate Evaluation of RESCUE and that of the corresponding base model. Specifically,

$$\text{Retention} = \frac{1}{20}\sum_{s=1}^{20}\frac{A_s^{\text{RESCUE}}}{A_s^{\text{Base}}} \times 100\%,$$

where A_s^{RESCUE} and A_s^{Base} denote Aggregate Evaluation in repair setting s for RESCUE and its corresponding base model. The 20 settings comprise two heterogeneous GSM8K repairs, ten independently tuned pattern-specific repairs, four re-localized rephrase repairs, two final models after sequential repair, and two MedMCQA repairs. Each model–setting pair is weighted equally. This calculation yields an average retention of 99.4% relative to the corresponding base models.

B.1 Detailed Results on Non-Target Benchmarks

Purpose and protocol. The heterogeneous GSM8K experiment asks whether one circuit can be localized from a mixture of failure types and then tuned without broadly degrading unrelated capabilities. Figure 2 compresses the eight non-target tasks into Aggregate Evaluation. Table 4 expands this average and reports the underlying benchmark values for the base model, direct LoRA, supervised-only localization, and the complete RESCUE pipeline. Supervised-only and RESCUE use their respective localized circuits for circuit-restricted tuning, while LoRA updates low-rank adapters without a localized circuit.

Ablation and circuit compactness. Before circuit-restricted tuning, suppressing the heterogeneous circuit improves full-set repair accuracy from 0% to 50.5% on Qwen3-8B and from 6.0% to 69% on Llama-3.1-8B-Instruct. The corresponding circuits contain only 1.49% and 1.40% of the eligible MLP rows, respectively. On the public GSM8K benchmark, ablation changes accuracy from 80% to 84% for Qwen3-8B and from 73% to 76% for Llama-3.1-8B-Instruct. These pre-tuning results establish that suppressing the localized rows already mitigates the mixed failure distribution.

Qwen3-8B. RESCUE exactly retains the base Aggregate Evaluation of 66.63%. Its individual changes relative to the base model are small and balanced: ARC-C and HellaSwag each increase by one point; WinoGrande and PIQA each decrease by one point; TruthfulQA and MMLU-Math remain unchanged; MMLU-Stats increases by one point; and BBH decreases by one point. LoRA obtains a slightly higher aggregate of 66.75%, but its benchmark profile is less uniform, including a four-point reduction on BBH that is offset by improvements on the two MMLU subsets. The supervised-only circuit reaches 65.00%, with its largest reduction occurring on BBH.

Llama-3.1-8B-Instruct. The capability-preservation differences are larger on Llama-3.1-8B-Instruct. RESCUE reaches 61.25%, compared with 62.38% for the base model, 60.00% for supervised-only localization, and 56.63% for LoRA. RESCUE exactly preserves ARC-C, HellaSwag, PIQA, and TruthfulQA, while changing WinoGrande and MMLU-Math by one point and MMLU-Stats by two points. Its largest reduction is five points on BBH. By comparison, LoRA decreases BBH from 73% to 38%, which accounts for much of its aggregate loss, while supervised-only reaches 62% on BBH.

Table 4: Detailed post-tuning accuracy (%) after heterogeneous GSM8K repair; Aggregate Eval. is the macro-average over the eight non-target benchmarks.

Model	Benchmark	Base	LoRA	Supervised-only	RESCUE
Qwen3-8B	ARC-C	56	57	57	57
	HellaSwag	75	76	75	76
	WinoGrande	68	67	68	67
	PIQA	78	77	78	77
	TruthfulQA	36	36	36	36
	MMLU-Math	50	53	55	50
	MMLU-Stats	72	74	74	73
	BBH	98	94	77	97
	Aggregate Eval.	66.63	66.75	65.00	66.63
Llama-3.1-8B-Instruct	ARC-C	56	51	53	56
	HellaSwag	80	78	79	80
	WinoGrande	74	74	73	73
	PIQA	81	80	81	81
	TruthfulQA	38	37	39	38
	MMLU-Math	42	43	39	41
	MMLU-Stats	55	52	54	53
	BBH	73	38	62	68
	Aggregate Eval.	62.38	56.63	60.00	61.25

Summary. Across both architectures, heterogeneous repair does not affect all non-target tasks uniformly. Nevertheless, RESCUE either matches the base aggregate or remains substantially closer to it than the alternative repair pipelines. Together with the higher GSM8K accuracy in the main paper, the detailed results support the conclusion that outcome-guided mask refinement and pruning improve targeted repair without requiring broad parameter updates.

B.2 Post-Tuning Results for Pattern-Specific Repair

Purpose and protocol. The main paper establishes a causal effect by ablating each pattern-specific circuit. Here we evaluate what happens after the same circuit is used as the trainable parameter subset. Each of the five circuits is tuned independently from the corresponding base model. We then evaluate the resulting model on GSM8K, MATH-500, and the eight non-target benchmarks. Table 5 reports all ten independently tuned models together with their respective base models.

GSM8K before and after tuning. The public GSM8K benchmark is evaluated both immediately after circuit ablation and after circuit-restricted tuning. In the Qwen3-8B pattern order shown in Table 1, the post-ablation scores are 80%, 82%, 80%, 82%, and 82%, compared with a base accuracy of 80%; the corresponding post-tuning scores are 80%, 81%, 80%, 86%, and 82%. For Llama-3.1-8B-Instruct, the post-ablation scores are 75%, 73%, 74%, 74%, and 74%, versus a 73% base accuracy, and the post-tuning scores are 73%, 75%, 75%, 74%, and 73%.

Qwen3-8B. Aggregate Evaluation ranges from 66.25% to 67.38%, compared with 66.63% for the base model. The largest positive difference is obtained after tuning the Dynamic State Tracking circuit ($+0.75$ points), whereas the largest reduction is produced by Declining Group Accumulation (-0.38 points). GSM8K remains at or above its 80% base value for every circuit and reaches 86% after tuning Declining Group Accumulation. MATH-500 remains within 74–77%, compared with 75% for the base model.

Table 5: Post-tuning accuracy (%) after independently repairing each pattern-specific GSM8K dataset; Aggregate Eval. is the macro-average over the eight non-target benchmarks.

Model	Benchmark	Base	Misleading Premise Back-solving	Dynamic State Tracking	Dropped-lowest Average	Declining Group Accumulation	Multiplicative Increment Back-solving
Qwen3-8B	GSM8K	80	80	81	80	86	82
	MATH-500	75	76	75	77	74	75
	ARC-C	56	56	56	57	55	58
	HellaSwag	75	75	75	75	75	75
	WinoGrande	68	69	68	68	67	68
	PIQA	78	77	78	77	77	77
	TruthfulQA	36	36	36	36	36	37
	MMLU-Math	50	51	52	51	52	51
	MMLU-Stats	72	72	77	73	70	74
	BBH	98	98	97	97	98	97
	Aggregate Eval.	66.63	66.75	67.38	66.75	66.25	67.13

Model	Benchmark	Base	Cumulative Target Gap	Scale Conversion	Percentage Risk	Unit Price and Total Cost	Fees and Change
Llama-3.1-8B-Instruct	GSM8K	73	73	75	75	74	73
	MATH-500	48	46	47	48	46	46
	ARC-C	56	55	55	56	56	56
	HellaSwag	80	80	80	79	79	80
	WinoGrande	74	74	73	74	74	73
	PIQA	81	81	81	82	81	81
	TruthfulQA	38	38	38	38	38	37
	MMLU-Math	42	40	42	43	42	40
	MMLU-Stats	55	55	55	54	53	52
	BBH	73	74	76	75	70	70
	Aggregate Eval.	62.38	62.13	62.50	62.63	61.63	61.13

Llama-3.1-8B-Instruct. Aggregate Evaluation ranges from 61.13% to 62.63%, versus 62.38% for the base model. Cumulative Target Gap and Scale Conversion remain within 0.25 and 0.12 points of the base aggregate, respectively, while Percentage Risk is 0.25 points higher. The two lowest aggregates occur for Unit Price and Total Cost and Fees and Change, driven mainly by lower BBH and MMLU-Stats results. GSM8K remains between 73% and 75%, and MATH-500 remains between 46% and 48%. Thus, the model-specific repair effect remains localized even for circuits whose ablation changes full-set repair accuracy by more than 50 percentage points.

Summary. Across the ten independently repaired models, the non-target aggregate differs from the corresponding base by at most 0.75 points for Qwen3-8B and 1.25 points for Llama-3.1-8B-Instruct. These results complement the ablation evidence in the main paper: the localized circuits are not only causally associated with the recurring failures, but can also serve as restricted update regions that correct those failures without inducing the broad shifts expected from unconstrained fine-tuning.

Layer-wise circuit organization. For layer ℓ, we define the enrichment score as $E_\ell = (k_\ell/n_\ell)/(K/N)$, where k_ℓ and n_ℓ are the selected and eligible rows in layer ℓ, and K and N are their model-wide totals. Figure 9 shows that Qwen3-8B circuits are relatively diffuse, whereas Llama-3.1-8B-Instruct circuits are enriched in middle-to-late layers. Specifically, 65.7% of the selected rows lie in the latter half of Llama-3.1-8B-Instruct, compared with 52.6% for Qwen3-8B, indicating model-specific depth organization.

B.3 Circuit Re-localization on Rephrased Inputs

Purpose and protocol. The main-text cross-rephrase experiment applies circuits localized from the original formulations directly to all five rephrased pattern sets per model. That experiment isolates transfer: it performs no re-localization, additional mask refinement, or circuit-restricted tuning on the rewritten inputs. As a complementary and more computationally intensive analysis, we rerun the complete RESCUE pipeline on two rephrased patterns per model. The evaluated Qwen3-8B patterns are Misleading Premise Back-solving and Dynamic State Tracking; the Llama-3.1-8B-Instruct patterns are Cumulative Target Gap and Percentage Risk. Table 6 compares direct circuit transfer with re-localization and reports the downstream post-tuning results.

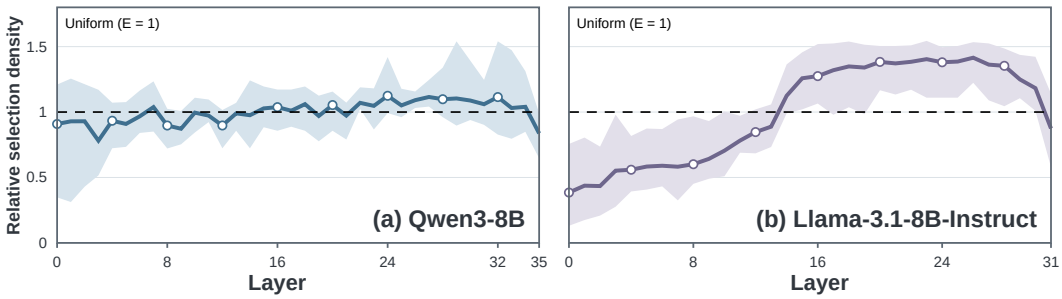


Figure 9: Layer-wise enrichment of localized error-circuit rows, normalized by each circuit's overall selection density. Curves and shading show the mean and min–max range across five pattern-specific circuits; $E = 1$ denotes no enrichment.

Ablation after re-localization. The re-localized circuits raise full-set accuracy to 94–96% across all four settings while selecting only 0.65–0.69% of the candidate MLP rows. On Qwen3-8B, direct transfer raises the two rephrased sets from 28% to 42% and from 17% to 25%, whereas re-localization reaches 95% on both. Re-localization therefore adds 53 and 70 points over direct transfer. On Llama-3.1-8B-Instruct, direct transfer is already stronger, reaching 60% for Cumulative Target Gap and 92% for Percentage Risk; re-localization raises these values to 94% and 96%, adding 34 and 4 points. The small four-point increment for Percentage Risk is not a failure of re-localization: the original circuit already transfers almost completely to that rephrased distribution.

Post-tuning preservation. For Qwen3-8B, the two re-localized circuits obtain Aggregate Evaluation scores of 66.75%, compared with 66.63% for the base model. GSM8K remains at 80–81%, and MATH-500 remains at 75–77%. For Llama-3.1-8B-Instruct, the aggregates are 61.63% and 62.00%, compared with 62.38% for the base model; GSM8K remains at 73–74%, while MATH-500 decreases from 48% to 44–45%. Across the eight aggregate components, the changes are small and do not exhibit a consistent collapse on any shared benchmark.

Interpretation and scope. Direct transfer and re-localization answer different questions. Direct transfer tests whether an originally localized mechanism remains causally relevant after surface reformulation, while re-localization tests whether RESCUE can rediscover a compact circuit when given the rewritten distribution itself. The results support both claims, but the re-localization analysis covers only two patterns per model and should be interpreted as a complementary robustness check rather than an exhaustive evaluation over every rephrased pattern.

B.4 Post-Tuning Results after Sequential Repair

Purpose and protocol. The main sequential-repair experiment evaluates whether errors can be repaired as they are discovered, without restarting from the original model or erasing earlier corrections. Beginning with the base model, each stage localizes and repairs the next pattern using the model state inherited from the preceding stage. The main-text heatmap evaluates all five pattern sets after every stage. Figure 10 complements this within-repair-set analysis by comparing the public-benchmark performance of the base model with the final model obtained after all five stages (AR5).

Qwen3-8B. After five stages, GSM8K and MATH-500 remain unchanged at 80% and 75%, respectively. Aggregate Evaluation changes only from 66.63% to 66.25%. At the benchmark level, HellaSwag and MMLU-Math each improve by two points, ARC-C, WinoGrande, PIQA, and TruthfulQA remain unchanged, MMLU-Stats decreases by three points, and BBH decreases by four points. The small aggregate difference therefore reflects offsetting benchmark-level changes rather than a uniform downward shift.

Llama-3.1-8B-Instruct. GSM8K increases from 73% to 76% after sequential repair, while MATH-500 decreases from 48% to 42%. Aggregate Evaluation decreases by two points, from

Table 6: Re-localization on rephrased pattern-specific datasets. Transferred Abl. applies the original circuit, whereas Re-localized Abl. uses a circuit localized on the rephrased data; the bottom panel reports post-tuning accuracy.

(a) Rephrased full-set ablation

Model		Misleading Premise Back-solving	Dynamic State Tracking
Qwen3-8B	Rephrased Base	28%	17%
	Transferred Abl.	42%	25%
	Re-localized Abl.	**95%**	**95%**
	Δ vs. Base (pp)	**+67**	**+78**
	Circuit	0.68%	0.69%
Model		**Cumulative Target Gap**	**Percentage Risk**
Llama-3.1-8B-Instruct	Rephrased Base	25%	26%
	Transferred Abl.	60%	92%
	Re-localized Abl.	**94%**	**96%**
	Δ vs. Base (pp)	**+69**	**+70**
	Circuit	0.65%	0.65%

(b) Post-tuning evaluation

Model		Base	Misleading Premise Back-solving	Dynamic State Tracking
Qwen3-8B	GSM8K	80	80	81
	MATH-500	75	75	77
	ARC-C	56	56	57
	HellaSwag	75	75	74
	WinoGrande	68	69	70
	PIQA	78	78	77
	TruthfulQA	36	36	37
	MMLU-Math	50	51	50
	MMLU-Stats	72	72	71
	BBH	98	97	98
	Aggregate Eval.	66.63	66.75	66.75
Model		**Base**	**Cumulative Target Gap**	**Percentage Risk**
Llama-3.1-8B-Instruct	GSM8K	73	74	73
	MATH-500	48	44	45
	ARC-C	56	55	56
	HellaSwag	80	80	79
	WinoGrande	74	73	74
	PIQA	81	81	81
	TruthfulQA	38	38	38
	MMLU-Math	42	41	41
	MMLU-Stats	55	53	53
	BBH	73	72	74
	Aggregate Eval.	62.38	61.63	62.00

62.38% to 60.38%. WinoGrande and MMLU-Math remain unchanged; PIQA and MMLU-Stats decrease by one point; HellaSwag decreases by two; ARC-C and TruthfulQA decrease by three; and BBH decreases by six. The larger change relative to Qwen3-8B shows that capability retention under repeated updates is architecture- and benchmark-dependent.



Figure 10: Benchmark accuracy for the base models and their final states after five sequential repairs (AR5).

Interpretation. The main-text heatmap shows that the final AR5 models retain high accuracy on the previously repaired pattern sets, with maximum later-stage decreases of only eight points. Figure 10 adds an important qualification: retaining prior repairs does not imply that every unrelated benchmark is exactly invariant. Nevertheless, five successive circuit-restricted updates preserve the Qwen3-8B aggregate almost completely and limit the Llama-3.1-8B-Instruct aggregate reduction to two points, while maintaining the accumulated repair effect. This supports sequential deployment while making the remaining model-dependent interference explicit.

B.5 Detailed Results for MedMCQA Repair

Purpose and protocol. The MedMCQA experiment tests whether the localization-and-repair procedure extends beyond grade-school mathematical reasoning. The main paper reports repair-set accuracy, the public MedMCQA benchmark, circuit size, and Aggregate Evaluation for the base model, LoRA, and RESCUE. Table 7 expands the aggregate into its eight constituent benchmarks and separately reports GSM8K as an additional cross-domain retention measure. The supervised-

Table 7: Detailed post-tuning accuracy (%) for MedMCQA repair. Aggregate Eval. averages the eight benchmarks from ARC-C through BBH and excludes GSM8K.

Model	Benchmark	Base	LoRA	RESCUE
Qwen3-8B	MedMCQA	59	**60**	**60**
	GSM8K	80	85	**86**
	ARC-C	56	57	**58**
	HellaSwag	75	**76**	**76**
	WinoGrande	68	68	**69**
	PIQA	78	76	**79**
	TruthfulQA	36	36	**37**
	MMLU-Math	**50**	49	**50**
	MMLU-Stats	**72**	69	71
	BBH	**98**	87	94
	Aggregate Eval.	66.63	64.75	**66.75**
Llama-3.1-8B-Instruct	MedMCQA	**59**	57	**59**
	GSM8K	**73**	71	72
	ARC-C	**56**	53	52
	HellaSwag	**80**	78	78
	WinoGrande	**74**	72	72
	PIQA	**81**	80	80
	TruthfulQA	**38**	36	35
	MMLU-Math	**42**	36	40
	MMLU-Stats	**55**	48	52
	BBH	**73**	68	69
	Aggregate Eval.	**62.38**	58.88	59.75

only baseline is not included in this setting because it was not run for the completed MedMCQA experiments.

Qwen3-8B. RESCUE and LoRA both obtain 60% on the MedMCQA benchmark, compared with 59% for the base model. Their non-target behavior differs substantially, however. RESCUE obtains an Aggregate Evaluation of 66.75%, slightly above the base model's 66.63%, while LoRA decreases to 64.75%. Relative to the base model, RESCUE improves ARC-C by two points and HellaSwag, WinoGrande, PIQA, and TruthfulQA by one point, preserves MMLU-Math, decreases MMLU-Stats by one point, and decreases BBH by four points. LoRA's lower aggregate is driven mainly by reductions of three points on MMLU-Stats and eleven points on BBH. GSM8K increases from 80% to 86% under RESCUE and to 85% under LoRA.

Llama-3.1-8B-Instruct. RESCUE preserves the base MedMCQA accuracy of 59%, whereas LoRA reaches 57%. Aggregate Evaluation decreases from 62.38% to 59.75% for RESCUE and to 58.88% for LoRA. Thus, RESCUE does not fully preserve the base aggregate on this model, but

remains 0.87 points above direct LoRA while achieving two points higher MedMCQA accuracy. The RESCUE changes relative to the base model range from one point on PIQA to four points on ARC-C and BBH. GSM8K changes only from 73% to 72%, compared with 71% under LoRA.