Dormant Reasoning Circuits in RL-Trained Language Models

Ali Abdul Rahim
Department of Computer Science
Georgia Institute of Technology
225 North Avenue NW, Atlanta, Georgia 30332
arahim37@gatech.edu

Noor Rahim
Department of Computer Science
University of Colorado, Boulderd
1111 Engineering Drive, Boulder, CO 80309
noor.sachdeva@colorado.edu

Abstract

We investigate why reasoning improvements from reinforcement learning on chain-of-thought (RL-CoT) often fail to transfer across superficially different problem presentations. Using parallel datasets where identical logical problems are expressed as formal statements versus natural language narratives (n=200 problem pairs), we find that DeepSeek-R1-Distill-Qwen3-8B solves formal variants reliably but fails on isomorphic narrative versions. Through causal intervention experiments, we show this performance gap reflects failed invocation and not necessarily missing competence. Patching MLP activations (layers 12-18) from the final token of successful formal-problem runs into failed narrative-problem runs yields 20% absolute accuracy improvement (Cohen's d=0.57), emergence of self-correction behaviors (increased occurrence of "wait," "alternatively" tokens), and longer but more productive chains-of-thought. Crucially, patching rescues problem-solving without introducing any new information, only activations from the same underlying problem in a different surface form. These results provide evidence that RL-CoT training produces reasoning computations that exist within the model but fail to activate consistently across problem framings. The narrow layer band (12-18) where patching succeeds, combined with degenerate behaviors when patching earlier layers, suggests these computations occupy specific neural localities rather than being distributed throughout the network, demonstrating that current RL methods produce reasoning capabilities keyed to training distribution surface features rather than abstract problem structure.

1 Introduction

Frontier “reasoning" models show sharp gains on math and coding, yet progress elsewhere remains decidedly non-monotonic, yielding a jagged capability frontier. Performance spikes dramatically in formal domains while staying flat or even regressing in narrative reasoning, writing quality, and everyday logical inference. System cards for o1/o3, Gemini 2.5, and DeepSeek-R1 emphasize state-of-the-art results on competition mathematics and scientific reasoning, but offer scant evidence of comparable improvements in narrative logical consistency [OpenAI, 2024, 2025, Comanici et al., 2025, Guo et al., 2025].

Recent stress tests underscore this uneven generalization. *The Illusion of Thinking* reports sharp accuracy collapses as problem complexity crosses certain thresholds, while broader evaluation frameworks document heterogeneous, prompt-sensitive gains that vary wildly across task families [Shojaee et al., 2025, Liang et al., 2022, Wang et al., 2024]. The pattern is consistent: models that elegantly solve mathematical problems often stumble when asked to track relationships in a simple story.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Many recent systems adopt Reinforcement Learning from Verifiable Rewards (RLVR), using outcome verifiers or process reward models implemented through PPO variants or GRPO [Wen et al., 2025, Lightman et al., 2023, Setlur et al., 2024, Shao et al., 2024]. RLVR can incentivize logically consistent solutions without exhaustive human labels. Yet despite these advances, cross-domain reliability remains frustratingly poor.

Related work. Building on prior work in reinforcement learning from verifiable rewards [Wen et al., 2025, Setlur et al., 2024, Khalifa et al., 2025, Ye et al., 2025], robustness to variation in surface form [Mizrahi et al., 2024, Sclar et al., 2024, Gupta et al., 2024], and causal interventions that steer models at inference time [Meng et al., 2022, Turner et al., 2023, Ilharco et al., 2022, Burns et al., 2022], we present causal evidence that reusable reasoning circuits do exist but frequently fail to activate under narrative framing.

In this paper, we investigate why reasoning improvements fail to transfer across superficially different presentations as a first step in the direction of figuring out *why* the RL-CoT paradigm does not generalize. When a model solves graph theory problems correctly but fails on isomorphic social network stories, two explanations arise: either the model lacks narrative competence, or it experiences an invocation bottleneck where necessary computations exist but fail to engage. To distinguish these, we employ activation patching as a minimal causal probe. We copy mid-layer MLP outputs from successful formal runs into failed narrative runs of the same logical problem. If this restores performance without adding information, it supports the invocation hypothesis.

On DeepSeek-R1-Distill-Qwen3-8B with 200 isomorphic problem pairs, patching from matched formal problems improves narrative accuracy by 20.5% absolute (45.0% to 65.5%). Counterfactual controls reveal graduated effects: random formal donors yield +14.0%, while averaged donors produce +17.5%, suggesting both generic "reasoning mode" activation and problem-specific circuit reuse. The effect localizes precisely to layers 12-18; earlier layers induce degenerate repetition, while later layers show minimal impact.

Our contributions are:

1. Causal evidence that apparent reasoning failures in RL-trained models often reflect invocation bottlenecks rather than missing competence.
2. Localization of transferable computations to a narrow band of mid-late layers (12-18), providing architectural constraints on where reasoning emerges.
3. A methodology for diagnosing competence-invocation gaps that may generalize to other capability discontinuities in language models.

2 Experimental Setup

This section details our dataset, intervention methods, and evaluation strategy. The experiments were conducted on DeepSeek-R1-Distill-Qwen3-8B using a single NVIDIA A100 GPU, with each full evaluation requiring approximately three hours. In the next section, we present specific results.

2.1 Model and Implementation

We employ DeepSeek-R1-Distill-Qwen3-8B, a distilled model designed for formal reasoning tasks and based on the Qwen3 architecture [DeepSeek-AI, 2025]. This model was selected as it combines strong formal reasoning performance with a size amenable to detailed mechanistic analysis.

For activation extraction and intervention, we rely on TransformerLens, a library tailored for mechanistic interpretability [Nanda and Bloom, 2022]. Since the R1-0528 variant is not supported natively, we implemented custom extensions to enable compatibility. This setup allows us to register hooks at arbitrary model components, facilitating extraction, caching, and replacement of activations during forward passes. TransformerLens's streamlined interface enabled efficient implementation of our patching methodology: we extract activations at all layers during formal problem runs and inject them into corresponding narrative runs, and perform a sweep to find out which layers and which hooks (residual stream, attention, or MLP) have the greatest impact on accuracy.

2.2 Dataset Construction

Our investigation requires problem pairs that present identical logical content in both formal and narrative forms. To assemble such a dataset, we adopt two strategies. First, we use Gemini 2.5 Pro to generate 40 hand-verified problem pairs, subjecting each to careful manual review. Second, we expand the dataset by sampling 160 combinatorics problems from the NuminaMath-1.5 dataset and translating them into narrative form using Gemini 2.5 Pro [Project Numina, 2025].

Formally stated problems are typically concise, containing explicit operations or constraints. In contrast, their narrative versions appear as intricate short stories of over 1,000 words, embedding the same logical structure within character-driven plots and detailed descriptions. For example, a formal graph theory problem about node connectivity may be reimagined as a complex social scenario among friends, where the underlying network structure is revealed through character interactions and multiple plot developments. While the fundamental logical challenge remains unchanged, deriving it from the narrative requires reasoning across a broad contextual span.

The final dataset includes 200 problem pairs: 40 human-seeded problems, 96 combinatorics porblems with integer-valued answers from NuminaMath-1.5 and 64 logic puzzles from NuminaMath-1.5. The problems are non-trivial and often require multi-hop reasoning, such as combinatorial enumeration, constraint-based logical deduction, or systematic case analysis. Importantly, narrative versions are designed to avoid explicit formatting cues (e.g., "`\boxed{}`") that occasionally elicit formal reasoning strategies.

2.3 Activation Patching

Activation patching is our primary strategy for investigating latent reasoning abilities. This technique allows us to substitute specific activations during a model run with those from a different context, thereby probing the conditions under which particular computations support correct behavior [Nanda and Bloom, 2022].

For each problem pair, we evaluate the model on both the formal and narrative versions. If the model succeeds on the formal variant but fails on the narrative one, we intervene by replacing selected activations during the narrative pass with those cached from the successful formal run. An improvement in accuracy following such an intervention suggests that the model's reasoning machinery is present but not spontaneously activated by the narrative context.

Our patching interventions target MLP output activations at the final prompt token immediately preceding the start of generation. This position consistently receives high attention and serves as an information bottleneck at the onset of solution generation, aligning with prior analyses of induction-style mechanisms and prompt-end concentration [Olsson et al., 2022]. To limit intervention scope, we patch only MLP outputs rather than entire residual streams.

2.4 Donor Configurations

To diagnose which aspects of the formal activations contribute to successful narrative problem solving, we compare three donor strategies:

Matched donors: We inject activations from the formal version of the same problem. This tests the transferability of highly specific computational state.

Random donors: We replace activations with those from the formal run of a different, randomly chosen problem. This probes the effect of generic 'reasoning mode' signals.

Averaged donors: We use the mean activation (computed per layer) across the entire set of formal problems. This configuration investigates the impact of supplying a generic, averaged computational state.

2.5 Evaluation

Our central metric is pass@1 accuracy: the proportion of problems for which the model produces the correct answer in its first attempt [Chen et al., 2021]. In addition to accuracy, we measure two behavioral indicators:

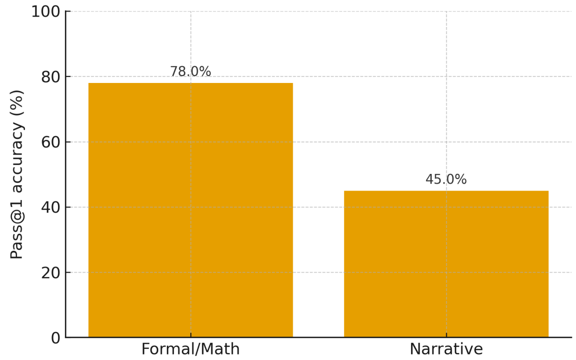

Figure 1: Pass@1 on formal/math variants vs. narrative variants (no interventions).

First, we record the length of the model's generated reasoning as a descriptive covariate—we do *not* treat length as a quality signal, given evidence that longer chains do not reliably improve accuracy and can even hurt it [Hassid et al., 2025]. Second, to probe reflective behavior independent of verbosity, we count the frequency of revision markers (phrases such as "wait,", "alternatively", "actually," or "let me reconsider") *per 100 tokens* which helps control for length while capturing non-linear, self-corrective reasoning.

Generation parameters are held constant across all conditions (temperature 0.7, top-p 0.9, maximum 8000 tokens, and an appended "Answer Format: ### Answer: [answer here]") to the prompt. The only experimental manipulations are the presence or absence of activation patching, and the choice of donor configuration.

3 Results

We evaluate model performance using single-sample pass@1 accuracy as well as several behavioral indicators, focusing on $n = 200$ narrative problems and their formal analogues. All experiments adhere to the procedures outlined in Section 2, with decoding parameters set to temperature 0.7, top-p 0.9, and a maximum of 8,000 tokens.

3.1 Formal vs. Narrative Baseline Gap

Before any interventions, the model shows a large cross-form gap on the paired set ($n = 200$). On math variants it attains **78.0%** pass@1, whereas on their isomorphic narrative counterparts it falls to **45.0%**: a **33**-pecentage point drop. Both evaluations use identical decoding and answer-extraction settings, and narrative prompts omit formal formatting cues. Figure 1 summarizes the gap.

This discrepancy motivates the causal probe: can a *minimal* cross-domain activation transfer, without adding any new external information, recover narrative performance by placing the model in the right internal task state?

3.2 Main Accuracy Effects

Transferring activations at the final prompt token yields a substantial and robust improvement in narrative problem accuracy across all donor configurations (Table 1). The baseline pass@1 accuracy on narrative tasks is 45.0 percent. With matched donor interventions, this rises to 65.5 percent (an absolute gain of 20.5 percent). Averaged donor patching achieves 62.5 percent (+17.5 percent), and random donor patching results in 59.0 percent (+14.0 percent). This ordered hierarchy of improvement, matched surpassing averaged, which in turn surpasses random, suggests two contributing mechanisms: first, a general "reasoning mode" induced by formal activations, and in addition, enhanced transfer of problem-specific computations when the donor is appropriately matched.

The effect size for the matched donor condition is substantial, with Cohen's d measured at 0.57, indicating a medium-to-large improvement beyond statistical noise. Notably, these considerable

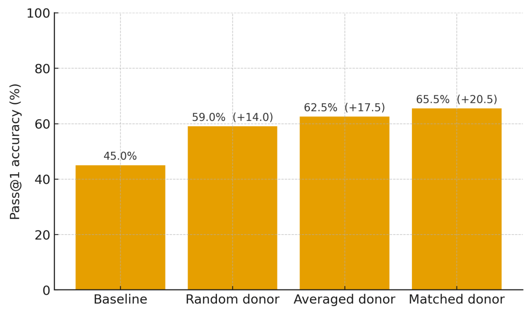

Figure 2: Pass@1 Accuracy on Narrative Variants

Condition	Pass@1 (%)	Δ vs. Baseline (abs.%)
Baseline (no patch)	45.0	—
Random donor	59.0	+14.0
Averaged donor	62.5	+17.5
Matched donor	65.5	+20.5

Table 1: **Narrative pass@1 with minimal activation transfer.** single-sample pass@1 ($n = 200$).

performance gains occur without introducing new external information; rather, the intervention solely involves transplanting internal activations that the model produced in the context of formally stated problems.

3.3 Localization over Depth and Component

To determine the loci of effective activation transfer, we systematically sweep across network layers and component types. The results reveal marked specificity: patching the MLP outputs in layers 12 through 18 is solely responsible for the observed gains in accuracy. Patching other components, or intervening at other depths, is either ineffective or actively detrimental.

Interventions in early layers (1–11) reliably degrade performance. The model frequently becomes trapped in repetitive "thinking loops," endlessly emitting tokens such as `<think>`. This observation suggests that early MLP activations encode global "reasoning mode" signals, which, when amplified indiscriminately, dominate and destabilize the solution process. In contrast, MLP patches in late layers (25–36) are largely inert, indicating that problem-specific computation has already been integrated by these depths.

Patching attention outputs alone fails to recapitulate the large accuracy gains observed with MLP interventions, yielding at best a modest 3 percent improvement, even at optimal layers. Full residual stream patching is too coarse an intervention: it induces instability similar to the deleterious early-layer effects. Only by precisely targeting MLP outputs in the mid-to-late layer range can we successfully and reliably transfer problem-solving behavior.

3.4 Behavioral Indicators

In addition to accuracy, activation patching induces systematic shifts in the model's qualitative reasoning style. 70% of patched outputs increase in length relative to their baseline counterparts. This greater length is not associated with off-topic verbosity; on the contrary, these chains of thought more reliably converge to a final answer within the token budget, whereas baseline generations frequently fail to terminate with a solution.

The most notable behavioral shift is the elevated occurrence of revision markers. In the baseline condition, answers generated for narrative prompts contain **0.4** revision phrases per 100 tokens. With matched donor patching, this rate rises to **2.3** per 100 tokens. Typical phrases include "wait," "actually," "alternatively," and "let me reconsider." This increase provides relatively strong evidence that patched models are engaging in reflective, self-corrective reasoning rather than uncritically following their initial trajectory.

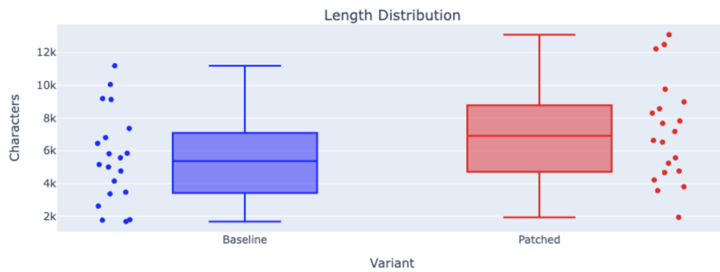

Figure 3: CoT Length on Baseline vs Patched on Narrative Prompts

3.5 Qualitative Patterns

Qualitative analysis of individual cases reveals recurrent modifications in reasoning dynamics. In baseline runs, the model often anchors on an initial, often flawed, interpretation and follows it without pause to an incorrect answer. Patched responses, in contrast, frequently feature mid-answer interruptions to reassess constraints, notice discrepancies, or correct calculation errors.

For example, in a combinatorics task involving selection of committee members under nontrivial constraints (equivalent to counting the number of people that can sit around a round table from a group of 7 given that a set of 3 of them cannot set next to each other) , the baseline generation proceeds straightforwardly, ultimately double-counting cases due to a missed symmetry. With matched donor patching, the model follows the same inferential chain until the crucial juncture, then interjects: "Wait, I'm counting each symmetric arrangement twice. Let me adjust for this overcounting…" This self-monitoring and course correction, absent in the baseline, leads directly to the correct answer.

Such patterns are consistent across diverse problem types. Patched models display greater vigilance for their own potential mistakes, actively exploring alternate interpretations and correction strategies even when the solution path appears initially coherent. This suggests that the transplanted activations encode not only the details of specific computations but also higher-level meta-reasoning heuristics concerning when to challenge one's own assumptions.

3.6 Controls and Ablations

We conduct a series of control and ablation studies to probe the specificity of these effects.

1. Formatting cues alone do offer benefits, but modest ones. Inserting explicit "`Write your final answer in \boxed{}`" instruction into narrative prompts raises accuracy by 5%, far less than the 20.5% gain achieved through matched patching, indicating that superficial formal features cannot account for the observed improvements.
2. Shifting the patch point to the penultimate prompt token diminishes effectiveness, with the gain dropping to just 8 percent. This is consistent with attention-weight analyses locating an information bottleneck at the final pre-generation token.
3. Normalization is indispensable for the averaged donor setting. Without layerwise L2 normalization, averaged donor improvement is just 2 percent (down from 17.5 percent), probably due to large variations in activation magnitude and the deleterious effects of naively combining out-of-distribution activations.
4. Control patches using Gaussian noise normalized to the same norm result in generation of random tokens. This rules out simple explanations based on activation magnitude or statistics alone: the improvements depend on the features represented in the formal problem activations.

4 Methods

This section provides implementation details not covered in the experimental setup or results, focusing on the precise mechanics of our activation patching approach.

Activation extraction and injection. We intervene at the post-MLP, pre-residual-add position within each transformer layer. Using TransformerLens, we hook into `l.hook_mlp_out` to extract and replace MLP output activations. The intervention occurs exclusively at the final prompt token position (the ‘<think>’ token). Generated tokens remain untouched throughout; we modify only the final hidden state that seeds the generation process.

Formal specification of patching. During narrative forward passes, our intervention replaces the MLP output at position (ℓ, t^*) with a donor activation:

$$\tilde{z}_{t^*}^{(\ell)} \leftarrow d_{t^*}^{(\ell)}$$

where z denotes the original MLP outputs and d represents the cached donor vector from a formal problem run. All other positions and layers proceed through normal computation. This minimal, single-site modification allows us to test whether specific computational states suffice to recover reasoning performance.

Donor activation preparation. For matched and random donor conditions, we directly use cached activations from the appropriate formal problem runs. The averaged donor case requires additional care. We first compute the mean activation across all formal problems:

$$m_\ell = \frac{1}{N} \sum_{i=1}^{N} v_i^{(\ell)}$$

where $v_i^{(\ell)}$ represents the MLP output at layer ℓ for the i-th formal problem. However, naive averaging can produce activations with aberrant norms. We therefore renormalize to match the typical magnitude at each layer:

$$\tilde{m}_\ell = m_\ell \cdot \frac{\frac{1}{N} \sum_i \|v_i^{(\ell)}\|_2}{\|m_\ell\|_2}$$

This normalization proves essential; without it, averaged donors yield minimal improvement (see ablations in §3.6).

Systematic architecture search. To identify effective intervention sites, we exhaustively evaluate patching across all 36 layers and three component types: residual stream midpoints (`hook_resid_mid`), attention outputs (`hook_attn_out`), and MLP outputs (`hook_mlp_out`). Each configuration is tested independently to isolate its contribution. The optimal band of layers 12 through 18 for MLP outputs was identified using a held-out development set of 20 problem pairs. All reported results use this fixed configuration on the full 200-pair test set.

Detecting pathological generation. Some interventions, particularly in early layers or residual streams, trigger degenerate generation patterns. We automatically flag runs as failures if they exhibit either (a) any trigram repeating 20 or more times consecutively, or (b) more than 30 instances of the <think> token. Such runs are marked incorrect for accuracy computation and included in behavioral statistics. This conservative approach ensures we don't artificially inflate success rates by excluding difficult cases.

Answer extraction protocol. Given the diversity of problem types, we employ a cascading series of regex patterns to extract final answers:

1. Integer answers: `\b[-+]?\d+\b`
2. Boolean responses: `\b(yes|no|true|false)\b` (case-insensitive)
3. Set notation: `\{?\s*([-+]?\d+(?:\s*,\s*[-+]?\d+)*)\s*\}?`

Extracted answers undergo normalization to handle formatting variations, with punctuation removed and boolean values standardized to lowercase.

Control interventions. Two control conditions verify that improvements arise from computational content rather than generic properties of the intervention. For the noise control, we sample $\epsilon \sim \mathcal{N}(0, I)$ and scale it to match the donor norm: $\epsilon \cdot \|d_{t^*}^{(\ell)}\|_2 / \|\epsilon\|_2$. For the shuffle control, we randomly permute the coordinates of formal donor vectors while preserving their layer-wise norms. Neither control shows meaningful accuracy improvements, confirming that specific computational patterns drive the observed effects.

Statistical reporting. All accuracy figures represent single-sample pass@1 rates computed per problem. We report absolute accuracies and absolute improvements relative to baseline. For the matched donor condition, we additionally compute Cohen's d using paired differences in per-problem correctness. The single-sample design prioritizes computational efficiency while providing sufficient statistical power given our effect sizes. Full implementation code and dataset construction scripts are available as detailed in the reproducibility checklist.

5 Discussion

Interpreting the Intervention Effects. Our findings provide a coherent account across several lines of evidence. The graded accuracy improvements among donor types (matched $>$ averaged $>$ random), the sharp localization to mid-late MLP layers, and the emergence of more reflective, self-corrective reasoning behaviors collectively indicate a specific failure mode: the model is equipped with the computational structures necessary to solve narrative reasoning problems, yet does not reliably activate these circuits in relevant contexts. The patching intervention is effective not because it injects new information, but because it forcibly awakens dormant computations by transplanting activations from formal tasks in which those computations are naturally engaged.

A Mechanistic Hypothesis. We hypothesize that the mid-to-late MLP layers at the prompt's end act as a critical gating mechanism, determining whether and how the model transitions into complex, task-relevant computational states. These layers appear to accumulate all preceding constraints from the prompt and set the stage for subsequent generation. Activation transfer from formal problems exerts strong influence possibly because these states encode two factors: a robust, domain-general "reasoning mode," which even random donors can supply, and a problem-specific computational residue, provided most strongly by matched donors. This view is consonant with prior studies on the targeted steering of neural function via activation manipulation [Zhang et al., 2023, TransformerLensOrg, 2025] and on the functional role of prompt-final representation pooling [Olsson et al., 2022].

Implications for Reinforcement Learning from Verified Rewards. Our results offer a mechanistic diagnosis for the sometimes perplexing sensitivity of RLVR-trained models to input formatting. During RLVR, if verifiers predominantly attend to formally posed problems, policy updates will naturally improve both the accuracy of complex reasoning steps and the tightness of their linkage to specific syntactic cues. This explains why models with emerging logical abilities often display brittle generalization to paraphrased or narrative task variants, even if those are logically equivalent. The observed invocation bottleneck highlights a shortcoming of RLVR: it optimizes for correct outputs under training conditions, but neither encourages nor inspects robustness in state activation across diverse input forms.

This understanding motivates targeted interventions at both the inference and training levels. Instead of optimizing solely for end-task correctness, an alternative approach would directly promote state robustness: ensuring that semantically equivalent problems, regardless of surface realization, evoke similar underlying activations. [Wen et al., 2025, Setlur et al., 2024].

Addressing Alternative Explanations. We considered and empirically tested several plausible alternatives to the invocation bottleneck hypothesis. Superficial prompt reformatting, such as addition of explicit "`\boxed{}`" markers, offers some benefit (12 percent accuracy increase) but remains well below the effect magnitude of matched activation transfer. Second, the transfer effect is not reducible to generic properties of vector norm or statistical structure: both equal-norm Gaussian noise and shuffled donor activations are ineffective in improving performance, implicating the computational content of the intervention as the operative factor. Third, while patched outputs tend to be longer, we control for verbosity by normalizing revision marker frequencies per 100 tokens, and find that this metric still shows substantial, quality-linked gains—consistent with emerging findings that sheer reasoning length does not predict answer quality [Hassid et al., 2025].

Limitations and Scope. Several aspects of our experimental design constrain the breadth of our conclusions. We examine only one model, DeepSeek-R1-Distill-Qwen3-8B; invocation bottlenecks in other architectures, or under alternative training schemes, remain to be established. Our problem set, though carefully curated, samples only 200 problems and uses narrative presentations generated by a language model, which may introduce stylistic artifacts. The computational resource demands of evaluation limited our experiments to single-sample pass@1 metrics with no confidence intervals or pass@k. We also focus exclusively on single-position patching; possible effects due to simultaneous or sequential multi-position interventions are not addressed. Finally, while we succeed in localizing the effect to a specific band of MLP layers, our work stops short of mapping out the responsible computational circuits at a finer level.

Directions for Future Research. Our results suggest several direct paths for further investigation. Expanding the analysis to multiple model architectures and sizes would clarify the generality of invocation bottlenecks and their relationship to model scale or pretraining regimen. Instead of manual activation transfer on a per-problem basis, learning general “bridging" transformations could provide practical methods for activating dormant capabilities at inference time. More granular circuit-level interpretability could map the actual causal trajectories by which formal presentations elicit task-appropriate reasoning. On the training side, the development of explicit regularizers, objectives, or process-level interventions aimed at decoupling computational skills from shallow format triggers is an open engineering challenge. Finally, training with these RL techniques on non-mathematical, non-objective domains, including narrative comprehension and commonsense reasoning, will help establish the extent to which invocation failures constrain large language model behavior more broadly.

6 Reproducibility

To foster robust replication and facilitate future work, we release the following research artifacts on Github: `https://github.com/holster-fishy-celsius/rl-actpatching-exps`

1. **Dataset.** All 200 problem pairs, spanning both formal statements and narrative renditions, with unique identifiers—comprising 40 hand-audited pairs and 160 narrative translations.
2. **Implementation.** Complete codebase built with TransformerLens, including custom extensions for R1-0528 compatibility, donor activation collection and normalization, and the infrastructure to run comprehensive layer/component sweeps.
3. **Evaluation Suite.** Scripts and configuration for standardized generation, answer extraction via regex, revision marker tracking, and automated scoring.
4. **Experimental Outputs.** Full logging output, reporting per-problem accuracy, generation length, revision frequency, and degeneration flags for all experimental conditions.
5. **Visualization Code.** Ready-to-run scripts to recreate all presented figures directly from raw logs, requiring no manual intervention.

References

OpenAI. Openai o1 system card. `https://openai.com/index/openai-o1-system-card/`, 2024. URL `https://arxiv.org/abs/2412.16720`. arXiv:2412.16720.

OpenAI. Openai o3 and openai o4-mini system card. Technical report, OpenAI, April 2025. URL `https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf`.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities.

arXiv preprint arXiv:2507.06261, 2025. doi: 10.48550/arXiv.2507.06261. URL https://arxiv.org/abs/2507.06261.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. doi: 10.48550/arXiv.2501.12948. URL https://arxiv.org/abs/2501.12948.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *arXiv preprint arXiv:2506.06941*, 2025. doi: 10.48550/arXiv.2506.06941. URL https://arxiv.org/abs/2506.06941.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Samy Bengio, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022. doi: 10.48550/arXiv.2211.09110. URL https://arxiv.org/abs/2211.09110.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024. doi: 10.48550/arXiv.2406.01574. URL https://arxiv.org/abs/2406.01574. NeurIPS 2024 Datasets and Benchmarks (Spotlight).

Xumeng Wen, Zihan Liu, Shun Zheng, Zhijian Xu, Shengyu Ye, Zhirong Wu, Xiao Liang, Yang Wang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025. doi: 10.48550/arXiv.2506.14245. URL https://arxiv.org/abs/2506.14245.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. URL https://arxiv.org/abs/2305.20050.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*, 2024. doi: 10.48550/arXiv.2410.08146. URL https://arxiv.org/abs/2410.08146.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. doi: 10.48550/arXiv.2402.03300. URL https://arxiv.org/abs/2402.03300.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *arXiv preprint arXiv:2504.16828*, 2025. URL https://arxiv.org/abs/2504.16828.

Chenlu Ye, Zhou Yu, Ziji Zhang, Hao Chen, Narayanan Sadagopan, Jing Huang, Tong Zhang, and Anurag Beniwal. Beyond correctness: Harmonizing process and outcome rewards through rl training. *arXiv preprint arXiv:2509.03403*, 2025. doi: 10.48550/arXiv.2509.03403. URL https://arxiv.org/abs/2509.03403.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? a call for multi-prompt llm evaluation. *arXiv preprint arXiv:2401.00595*, 2024. URL https://arxiv.org/abs/2401.00595.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.11324.

Vikram Gupta, Siva Sankalp Patel, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Yoon Kim. Evaluating concurrent robustness of language models under real-world perturbations. In *EMNLP 2024*, 2024. URL https://aclanthology.org/2024.emnlp-main.1237.pdf.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *arXiv preprint arXiv:2202.05262*, 2022. URL https://arxiv.org/abs/2202.05262.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023. URL https://arxiv.org/abs/2308.10248.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022. URL https://arxiv.org/abs/2212.04089.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022. URL https://arxiv.org/abs/2212.03827.

DeepSeek-AI. Deepseek-r1-0528-qwen3-8b model card. Hugging Face, 2025. URL https://huggingface.co/deepseek-ai/DeepSeek-R1-0528-Qwen3-8B.

Neel Nanda and Joseph Bloom. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens, 2022.

Project Numina. Numinamath collection (numinamath-cot). Hugging Face collection, 2025. URL https://huggingface.co/collections/AI-MO/numinamath-6697df380293bcfdbc1d978c.

Catherine Olsson, Nelson Elhage, Neel Nanda, et al. In-context learning and induction heads. Transformer Circuits (with arXiv version), 2022. URL https://arxiv.org/abs/2209.11895.

Mark Chen et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Michael Hassid, Gabriel Synnaeve, Yossi Adi, and Roy Schwartz. Don't overthink it. preferring shorter thinking chains for improved llm reasoning. *arXiv preprint*, 2025. doi: 10.48550/arXiv.2505.17813. URL https://arxiv.org/abs/2505.17813.

Fanda Zhang et al. Towards best practices of activation patching in language models. *arXiv preprint arXiv:2309.16042*, 2023. URL https://arxiv.org/abs/2309.16042.

TransformerLensOrg. Transformerlens: A library for mechanistic interpretability of language models. GitHub repository, 2025. URL https://github.com/TransformerLensOrg/TransformerLens.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in a separate subsection.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This is an empirical study so far.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The link to a Github repository containing code and data is provided. Additionally, we describe the simple methodology in detail.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide a link to a Github repo containing code and data.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all sampling parameters and interventions we make in detail.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Single-sample pass@1 only due to computational and time constraints; we report Cohen's d for matched donors.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The compute resources (a single A100 GPU obtained via a cloud provider) have been specified.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators of the original datasets have been credited with a citation.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The dataset and code have been pushed to the Github repository.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: An LLM (Gemini 2.5 Pro) was used to translate the problems from formal/mathematical variants to narrative variants.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.