TOWARDS GENERALIZABLE REASONING: GROUP CAUSAL COUNTERFACTUAL POLICY OPTIMIZATION FOR LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) excel at complex tasks with advances in reasoning capabilities. However, existing reward mechanisms remain tightly coupled to final correctness and pay little attention to the underlying reasoning process: trajectories with sound reasoning but wrong answers receive low credit, while lucky guesses with flawed logic may be highly rewarded, affecting reasoning generalization. From a causal perspective, we interpret multi-candidate reasoning for a fixed question as a family of counterfactual experiments. Building on this, we propose **G**roup **C**ausal **C**ounterfactual **P**olicy **O**ptimization (**GC^{2}PO**) to explicitly train LLMs to learn generalizable reasoning patterns. It proposes an episodic causal counterfactual reward that jointly captures (i) robustness, encouraging the answer distribution induced by a reasoning step to remain stable under counterfactual perturbations; and (ii) effectiveness, enforcing sufficient variability so that the learned reasoning strategy can transfer across questions. We then construct token-level advantages from this reward and optimize the policy, encouraging LLMs to favor reasoning patterns that are process-valid and counterfactually robust. Extensive experiments on diverse benchmarks demonstrate its advantages.

1 INTRODUCTION

Large Language Models (LLMs) have evolved from handling basic natural language processing tasks to tackling complex reasoning problems (Jimenez et al., 2024; Zhou et al., 2025; Sadik & Govind, 2025). While pre-training establishes a foundation, realizing their full potential, particularly in specialized reasoning domains, requires targeted post-training adjustments (Tie et al., 2025). Reinforcement Learning (RL) has emerged as a dominant paradigm to enhance reasoning capabilities (Shao et al., 2024; Zhang et al., 2024; Wang et al., 2025c). Among RL-based post-training approaches, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and its variants (Zhang et al., 2025a; Gu et al., 2025; Wang et al., 2025b) have garnered significant attention by reducing computational overhead while delivering substantial performance gains.

Despite these advancements, our empirical analysis reveals that existing methods may struggle to achieve generalizable reasoning: the models frequently overfit to in-distribution benchmarks and fail to answer unseen questions correctly, e.g., rephrasings, distractors, or harder variants (**Figure 1**). Through a toy experiment (**Subsection 2.2**), we trace this generalization failure to a misalignment in reward supervision, i.e., current algorithms heavily reward the final answer rather than the fidelity of the reasoning process. It inadvertently incentivizes LLMs to mimic answers and exploit superficial shortcuts rather than learning transferable reasoning principles. Specifically, current approaches primarily rely on outcome-based binary rewards (Shao et al., 2024; Yeo et al., 2025; Yang et al., 2025), assigning 1 to correct answers and 0 otherwise, regardless of the logical validity of the intermediate steps. Although some works use Process Reward Models (PRMs) (Wang et al., 2025b; Zhang et al., 2025c; Tan et al., 2025) to score intermediate steps, they are typically trained on chain-of-thought (CoT) traces filtered by final correctness, while abundant incorrect samples are treated as uniformly negative (Chen et al., 2025b; Ye et al., 2025a). Thus, they may inherit the outcome bias: "lucky guesses" driven by flawed logic receive undue reinforcement, whereas rigorous reasoning trails derailed by minor terminal slips are blindly penalized. To address this issue, we aim to answer a core question: *How can reward signals distinguish genuine reasoning from lucky guesses?*

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.

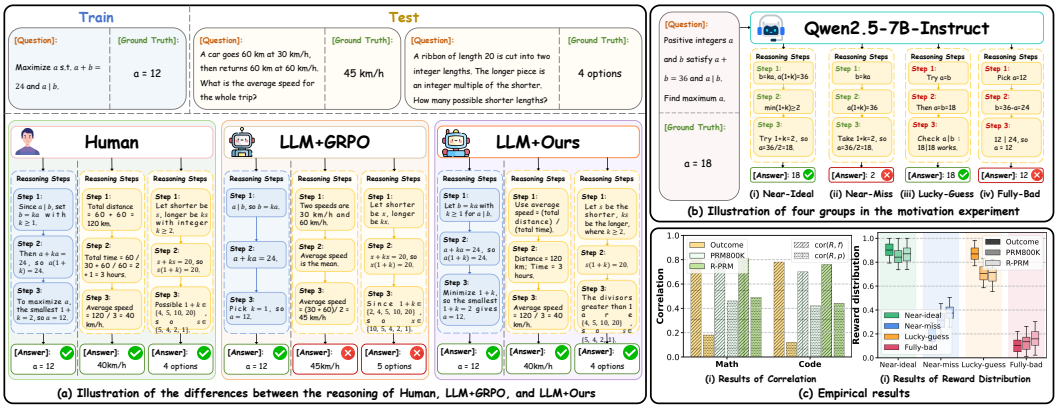


Figure 1: (a) Example of reasoning trajectories. (b) Examples of four trajectory groups (**Subsection 2.2**). (c) Empirical results of the motivation experiment. See **Appendix D.4** for details.

From a causal perspective (Pearl, 2009), we propose that this distinction hinges on causal invariance: genuine reasoning captures stable and transferable mechanisms that persist across context shifts, whereas lucky guesses exploit spurious correlations that happen to yield correct answers only by chance. As illustrated in **Figure 2**, generalizable reasoning relies on invariant structures (e.g., core constraints, correct variable binding, etc.) that remain valid across diverse problem formulations, rather than spurious cues (e.g., seeing "average" and mechanically applying a fixed formula) that only correlate with success in a narrow distribution (**Subsection 2.3**). Crucially, within the GRPO paradigm, the multiple trajectories sampled per prompt naturally provide a testbed of counterfactual trials (**Theorem 2.1**). By evaluating process behaviors across these parallel rollouts, we can isolate logically consistent transitions that signify genuine causal invariance. Motivated by this, we propose two complementary principles for reward design: (i) robustness, which assesses whether an intermediate step maintains stability under local perturbations, verifying the presence of an invariant mechanism; and (ii) effectiveness, which ensures the step retains sufficient task-relevant information, preventing the representations from collapsing into trivial states. Together, they establish a process-level supervisory signal that reliably credits sound reasoning while penalizing lucky guesses, even when final answers mislead, thereby steering LLMs toward robust reasoning generalization.

Building on these insights, we propose Group Causal Counterfactual Policy Optimization (GC^2PO). It introduces an episodic causal counterfactual reward to explicitly evaluate intermediate reasoning. This reward couples a stability term measuring causal invariance under perturbation with an expressiveness term that penalizes uninformative representations via optimal decay rates. GC^2PO operates in three steps: (i) Episode segmentation (**Subsection 3.1**): we automatically break down solution paths into several reasoning episodes; (ii) Reward calculation (**Subsection 3.2**): for each episode, we construct local perturbations in the representation space and calculate the reward with a Monte Carlo estimator, jointly estimating robustness and effectiveness; (iii) Policy optimization (**Subsection 3.3**): we combine episodic rewards with outcome rewards to allocate token-level advantages, encouraging the LLMs towards generalizable reasoning. Extensive experiments demonstrate its advantages.

The main contributions are as follows: (i) We reveal a structural bias in existing post-training through empirical and causal analyses, demonstrating how current reward mechanisms entangle process validity with final correctness and hinder generalization. (ii) We propose GC^2PO, a novel post-training framework that utilizes episodic causal counterfactual rewards to evaluate the robustness and effectiveness of underlying reasoning patterns. (iii) Extensive experiments across various benchmarks and models demonstrate the superiority of GC^2PO, improving the generalization of LLM reasoning.

2 Problem Settings and Analyses

2.1 Problem Settings

Our goal is to fine-tune a policy π_θ (i.e., LLM) to answer arbitrary questions. Taken GRPO (Shao et al., 2024) as an example: for each question $x \sim \mathcal{D}$, we sample K candidate trajectories $\{y_k\}_{k=1}^{K} \sim$

$\pi_{\theta_{\text{old}}}(\cdot|x)$ using the old policy $\pi_{\theta_{\text{old}}}$. For each y_k, we compute the outcome reward $R_{\text{out}}(x, y_k) \in \{0, 1\}$ using an automatic verifier (e.g., correctness plus formatting). Next, we calculate:

$$A_k(x) = \frac{r_k - \bar{r}(x)}{\sqrt{s_r^2(x)}}, \text{s.t. } \bar{r}(x) = \frac{1}{K}\sum_{k=1}^{K} r_k,\ s_r^2(x) = \frac{1}{K}\sum_{k=1}^{K}\big(r_k - \bar{r}(x)\big)^2, \tag{1}$$

By distributing this advantage uniformly over the tokens of y_k, we then optimize π_θ by maximizing:

$$\begin{aligned}
&\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\,\{y_k\}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\Big[\frac{1}{K}\sum_{k=1}^{K}\frac{1}{T_k}\sum_{t=1}^{T_k}\min\Big(\rho_{k,t}(\theta)A_k(x), \tilde{\rho}_{k,t}(\theta)A_k(x)\Big) - \beta_{\text{KL}}\,\mu_{\text{KL}}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big)\Big],\\
&\text{s.t. } \rho_{k,t}(\theta) = \frac{\pi_\theta\big(y_{k,t}|x, y_{k,<t}\big)}{\pi_{\theta_{\text{old}}}\big(y_{k,t}|x, y_{k,<t}\big)},\ \tilde{\rho}_{k,t}(\theta) = \text{clip}\,\big(\rho_{k,t}(\theta), 1-\epsilon, 1+\epsilon\big),\\
&\mu_{\text{KL}}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big) = \frac{\pi_{\text{ref}}(y_{k,t}|x, y_{k,<t})}{\pi_\theta(y_{k,t}|x, y_{k,<t})} - \log\frac{\pi_{\text{ref}}(y_{k,t}|x, y_{k,<t})}{\pi_\theta(y_{k,t}|x, y_{k,<t})} - 1,
\end{aligned} \tag{2}$$

where ϵ is the clipping radius, β_{KL} controls KL regularization μ_{KL}, and π_{ref} is the reference policy.

2.2 Empirical Evidence

Despite strong performance, existing reward mechanisms are tightly coupled to final correctness rather than the reasoning process (Shao et al., 2024; Khalifa et al., 2025; Tan et al., 2025), affecting generalization. To validate this, we conduct an experiment for analysis. Specifically, we construct an evaluation set from GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), and HumanEval (Chen et al., 2021), and use Qwen2.5-7B-Instruct to generate multiple trajectories per question. For each trajectory y_k, we record: (i) final correctness $f_k \in \{0, 1\}$, determined by symbolic solver or code execution; and (ii) process validity $p_k \in [0, 1]$, obtained from independent annotations of intermediate steps (human raters assisted by arithmetic checkers and code execution). We split the trajectories into four groups (**Figure 1(b)**) by evaluating whether their reasoning process and answer are correct, i.e., near ideal, near-miss, lucky guess, and fully bad. We then apply two families of reward mechanisms: (i) the outcome reward in GRPO (Shao et al., 2024); and (ii) representative PRMs, i.e., Qwen2.5-Math-7B-PRM800K and R-PRM-7B. We evaluate: (i) the correlation $\text{cor}(R(\cdot), f)$ and $\text{cor}(R(\cdot), p)$; and (ii) the rewards of different models over four groups.

As shown in **Figure 1(c)**, (i) existing methods align much more strongly with final correctness than with process validity: $\text{cor}(R(\cdot), f)$ is much higher than $\text{cor}(R(\cdot), p)$; (ii) the near-miss trajectories receive rewards close to fully bad trajectories, whereas lucky guesses obtain rewards comparable to truly high-quality solutions. These results underscore the limitations of existing reward mechanisms, which remain dominated by outcome signals and pay insufficient attention to the effectiveness and robustness of reasoning patterns. More details and analyses are provided in **Appendix E.1**.

2.3 Motivation Analysis

To address this limitation, in this subsection, we discuss the key question: how can we design fine-grained rewards that decouple process validity from final correctness, so as to encourage the LLMs to learn generalizable reasoning strategies rather than merely mimicking answers.

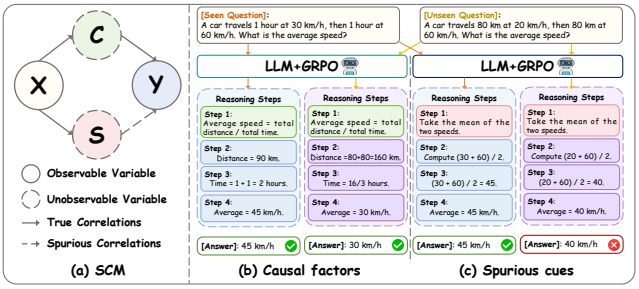


Figure 2: SCM of LLM reasoning.

We adopt a causal perspective and conduct a Structural Causal Model (SCM) for analyses (**Figure 2**). Following (Pearl, 2009), we posit that for a given question x, the reasoning behavior of an LLM is jointly driven by two latent factors: (i) the causal factors c, which encodes task-governing mechanisms that remain valid under distribution shifts, e.g., arithmetic rules, logical consistency, and valid variable binding; and (ii) spurious cues s, which encapsulates brittle, dataset-specific shortcuts that coincide with correct answers only within a narrow context, e.g., reliance on specific formatting templates or shallow keyword associations. Achieving generalizable reasoning necessitates that the LLM relies on the invariant structure c rather than overfitting to the spurious correlations s. However,

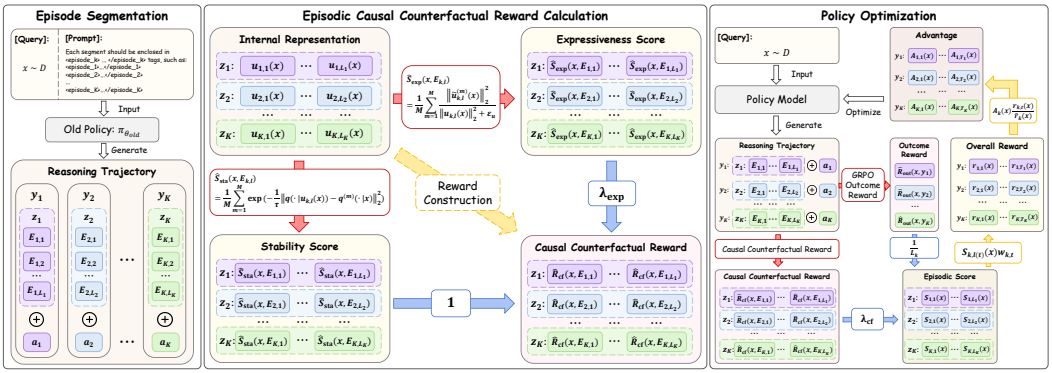


Figure 3: Illustration of GC^{2}PO. It segments reasoning into episodes (**Left**), calculates causal counterfactual reward (**Middle**), and optimizes LLMs by building token-level advantages (**Right**).

a challenge arises from the observational equivalence of existing reward mechanisms: a shortcut driven by s and a rigorous logical deduction driven by c yield the same correct final answer y; the reward mechanisms that rely solely on verifying y cannot distinguish between the two, frequently assigning equal credit to both of them. It may encourage shortcut behavior through $s \to y$. To mitigate this, we aim to design a reward that assigns higher scores with c while filtering out s, encouraging the learning of generalizable reasoning. See **Appendix D.5** for details.

To distinguish the causal factors c from spurious cues s, we propose to leverage the sampling mechanism inherent in GRPO to approximate a causal analysis. Instead of evaluating a single trajectory in isolation, we view the generation of multiple candidates as a set of counterfactual experiments. Formally, we establish the theoretical support as follows:

Theorem 2.1. *For a fixed input x, the reasoning of policy π_θ forms a Markov decision process (MDP) with transition kernel $P(s_{t+1}|s_t, a_t)$, satisfying: (i) There exists an SCM $\mathcal{M}$, without interventions, the trajectory distribution generated by $\mathcal{M}$ coincides with that of the MDP. (ii) For any alternative policy π'_θ or intervention on $P(s_{t+1}|s_t, a_t)$, the counterfactual trajectory distribution in $\mathcal{M}$ coincides with the distribution obtained by simulating the modified MDP.*

Theorem 2.1 is a specialization of the main results in (Oberst & Sontag, 2019; Ness et al., 2019), which show that finite MDPs can be represented by SCMs where counterfactual distributions match the dynamics of the modified process. For a question x, the sampled trajectories in GRPO can be viewed as parallel counterfactual experiments under shared exogenous noise but different reasoning paths. This perspective shifts the focus from identifying a single successful outcome to evaluating the structural stability of the reasoning process itself. By analyzing the variations across these parallel candidates, we can identify reasoning nodes that maintain logical consistency under perturbations, distinguishing them from those that are brittle or accidental. See **Appendix B.1** for proofs.

Guided by this perspective, we establish two core principles for reward design: robustness and effectiveness. Robustness measures the stability of the answer distribution induced by a given reasoning step under local perturbations. A high robustness score indicates that the step relies on invariant logic c rather than brittle spurious cues. However, robustness alone can favor degenerate or uninformative steps that remain trivially stable. We therefore introduce effectiveness. It measures the step's information content to ensure it conveys sufficient task-relevant information and prevents representation collapse. By integrating them, we move beyond binary outcome supervision. We reward sound reasoning even when the final answer is wrong, while penalizing trajectories that reach the right answer via flawed or spurious paths, thereby enforcing generalizable reasoning.

3 METHOD

Based on these analyses, we propose Group Causal Counterfactual Policy Optimization (GC^{2}PO). The core innovation is an episodic causal counterfactual reward. Instead of merely rewarding a trajectory that happens to be correct (which may rely on spurious shortcuts), it evaluates whether the underlying reasoning pattern is truly generalizable. GC^{2}PO operates in three key steps. First, we automatically

segment each question-solution pair into distinct episodes via prompt design (**Subsection 3.1**). Then, we compute a causal counterfactual reward for each episode. This reward consists of two terms that jointly measure the robustness and effectiveness of the reasoning step (**Subsection 3.2**). Finally, we leverage these rewards with the outcome reward to construct token-level advantages, optimizing the LLM to internalize generalizable reasoning strategies (**Subsection 3.3**). The framework is shown in **Figure 3**. The list of notations and pseudo-code are provided in **Appendix A** and **Appendix C**.

3.1 Episode Segmentation

Reasoning tasks vary significantly in complexity. However, existing GRPO-based methods typically treat the entire solution as a single episode (Wiering & Van Otterlo, 2012; Szepesvári, 2022). This coarse granularity is insufficient for complex reasoning: it ignores the internal logical structure and fails to identify which specific step caused a success or failure. To enable fine-grained evaluation, we reformulate the reasoning process as a sequence of distinct episodes, where each episode represents a coherent logical step. Formally, given a query $x \sim \mathcal{D}$, the policy $\pi_{\theta_{\text{old}}}$ generates a group of K candidate trajectories $\{y_k\}_{k=1}^{K}$. Each trajectory y_k consists of a reasoning chain z_k and a final answer a_k. We decompose the reasoning chain z_k into L_k discrete episodes, denoted as $z_k = \{E_{k,l}\}_{l=1}^{L_k}$, where each $E_{k,l}$ corresponds to the tokens of the l-th reasoning step. In practice, we enforce this segmentation via prompt engineering. The model is instructed to delimit reasoning steps using specific tags (e.g., `<episode_i>` and `</episode_i>`, see **Appendix E.5**), ensuring that each $E_{k,l}$ forms a self-contained semantic unit. It allows us to assign rewards to individual logical steps, enabling the model to learn from correct intermediate steps even if the final answer is incorrect.

3.2 Causal Counterfactual Reward Calculation

To determine whether an episode encodes a generalizable reasoning pattern, we propose an episodic causal counterfactual reward. The reward consists of: (i) a *stability term*: for robustness, it measures how much of the reasoning behavior of the episode survives under latent perturbations, and (ii) an *expressiveness term*: for effectiveness, it constrains the decay rate of the episode's representation, ensuring that the reasoning step retains sufficient information to be effective across different contexts. By optimizing for both, the LLM is incentivized to converge toward reasoning patterns that are both robust and effective. In this subsection, we first clarify the roles of the two terms and then formally define the reward (**Theorem 3.1**). Next, we show how to compute it efficiently with theoretical guarantees (**Theorem 3.2**). Finally, we discuss why this reward is effective.

To simulate potential variations in the reasoning context, we operate within the continuous representation space. It offers a more stable and computationally efficient alternative to perturbing discrete input tokens or model parameters (Wang et al., 2025b; 2024). For a given question x and episode $E_{k,l}$, we map the reasoning step to a latent vector $u_{k,l}(x, E_{k,<l} \in \mathbb{R}^d$ (simpled write as $u_{k,l}(x))$, which is derived from the hidden state of the last token. This vector determines the model's answer distribution $q(\cdot|u_{k,l}(x))$ over the answer space $\mathcal{A}$. We then apply a family of perturbation operators $\{g_m\}_{m=1}^{M}$ to generate perturbed representations $\tilde{u}_{k,l}^{(m)}(x) = g_m(u_{k,l}(x))$ and their induced answer distributions $q(\cdot|\tilde{u}_{k,l}^{(m)}(x))$. Intuitively, $u_{k,l}(x)$ encodes "what this episode is trying to do", while $\tilde{u}_{k,l}^{(m)}(x)$ simulates nearby "what-if" scenarios. The stability term quantifies the divergence between $q(\cdot|u_{k,l}(x))$ and $q(\cdot|\tilde{u}_{k,l}^{(m)}(x))$: episodes whose logic truly works under variations keep $q(\cdot|u_{k,l}(x))$ and $q(\cdot|\tilde{u}_{k,l}^{(m)}(x))$ close and thus receive high scores; brittle episodes exhibit large shifts and are penalized. The *expressiveness term* is defined from the viewpoint of the optimal decay rate (Cazenave & Haraux, 1998; Pazy, 2012): it tracks how fast the representation $u_{k,l}(x)$ decays along perturbation directions. If the perturbations quickly drive $u_{k,l}(x)$ toward a low-norm, uninformative state, the episode is structurally fragile and gets a low score; if the representation remains strong and decays slowly, the episode is rewarded. Together, they encourage episode representations that are both stable in their predictions and persistent in representation space.

We then present the formal definition of the episodic causal counterfactual reward with causal identifiability guarantees as follows. More analyses are provided in **Appendix D.3**.

Theorem 3.1. *For a fixed x and episode $E_{k,l}$, let $u_{k,l}(x) \in \mathbb{R}^d$ denote the episode representation and $q(\cdot|u_{k,l}(x)) \in \Delta(\mathcal{A})$ the induced answer distribution. Let $\{g_m\}_{m=1}^{M}$ be a family of perturbation*

operators acting on representations, and define $\tilde{u}_{k,l}^{(m)}(x) = g_m(u_{k,l}(x))$ *and* $q^{(m)}(\cdot|x) = q(\cdot|\tilde{u}_{k,l}^{(m)}(x))$. *Let* π_θ^* *be any policy that achieves the highest causal counterfactual reward* $R_{\text{cf}}^\star(x, E_{k,l})$*:*

$$R_{\text{cf}}^\star(x, E_{k,l}) = S_{\text{sta}}(x, E_{k,l}) + \lambda_{\text{exp}}\, S_{\text{exp}}(x, E_{k,l}),$$

$$\textit{s.t. } S_{\text{sta}}(x, E_{k,l}) = \mathbb{E}_m\Big[\exp\Big(-\tfrac{1}{\tau}\big\|q(\cdot|u_{k,l}(x)) - q^{(m)}(\cdot|x)\big\|_2^2\Big)\Big],\ S_{\text{exp}}(x, E_{k,l}) = \mathbb{E}_m\left[\frac{\big\|\tilde{u}_{k,l}^{(m)}(x)\big\|_2^2}{\big\|u_{k,l}(x)\big\|_2^2 + \varepsilon_u}\right], \tag{3}$$

where λ_{exp} *is weighting coefficient,* τ *and* ε_u *are hyperparameters following (Pazy, 2012). Then,* π_θ^* *block-identifies the true causal variables in the sense of* ***Definition B.3****.*

Intuition and Discussion. A larger $R_{\text{cf}}^\star$ indicates a more valuable episode for optimization. S_{sta} is maximized when $q(\cdot|u_{k,l}(x)) = q^{(m)}(\cdot|x)$ for all m, and decays exponentially as these distributions diverge; it therefore measures distributional robustness under pertubations. $S_{\text{exp}}(x, E_{k,l})$ captures how well the representation avoids collapsing under perturbations: with a local linearization $\tilde{u}_{k,l}^{(m)}(x) \approx A_m u_{k,l}(x)$, the ratio estimates the squared gain of A_m along $u_{k,l}(x)$, and its expectation reflects the effective energy decay. Larger S_{exp} thus implies weaker decay and more informative representations. Together, these two terms guarantee that the learned representation is invariant to nuisances while fully retaining essential causal information. By the inverse mapping theorem, this establishes an invertible correspondence between $u_{k,l}(x)$ and the underlying true causal variables, formally achieving block-identifiability. See **Appendix B.2** for more details.

Obtaining **Theorem 3.1**, we illustrate how to calculate it in practice. For each x, we run a forward pass to obtain $u_{k,l}(x)$ and $q(\cdot|u_{k,l}(x))$ with **Subsection 3.1**, then sample M perturbed representations $\tilde{u}_{k,l}^{(m)}(x)$. The implementation of the latent-space perturbation operators $\{g_m\}_{m=1}^M$ follows (Pearl, 2009; Kusner et al., 2017), satisfying the minimal change principle for counterfactuals. Next, we compute the answer distributions $q^{(m)}(\cdot|x)$ and norms, and substitute them into Eq.3 for S_{sta} and S_{exp}. Since computing $S_{\text{sta}}(x, E_{k,l})$ and $S_{\text{exp}}(x, E_{k,l})$ requires expectations over the full perturbation distribution, which are intractable in closed form, we approximate them with Monte Carlo estimators on M samples. Below, we establish:

Theorem 3.2. *Let the latent manifold be* $\mathcal{Z} = \mathcal{C} \times \mathcal{S}$ *endowed with a Riemannian metric, assume the perturbations* $\{g_m\}_{m=1}^M$ *locally span the tangent space of the non-causal manifold* $T_s\mathcal{S}$*. If a learned* u^* *satisfies* $S_{\text{sta}} \geq 1 - \epsilon$ *and* $S_{\text{exp}} \geq \gamma$*, then, maximizing* $R_{\text{cf}}^\star$ *minimizes the error bound to the oracle policy* $\pi_{\text{opt}}(\cdot|c) := \mathbb{E}_s[\pi^*(\cdot|c, s)]$ *with* $\mathbb{E}_s[D_{\text{KL}}(\pi_{\text{opt}}(\cdot|c) \,\|\, \pi^*(\cdot|c, s))] \leq \mathcal{O}(\sqrt{\epsilon})$.

Theorem 3.2 establishes that maximizing our proposed reward drives the learned policy to converge toward the optimal causal oracle π_{opt}. See **Appendix B.3** for proofs.

Finally, we explain why the reward is effective. It operates at the episode level and evaluates how the encoded reasoning generalizes, rather than only final correctness. The episodes containing genuinely valid reasoning steps tend to induce answer distributions that remain stable under mechanism-preserving perturbations, yielding high stability scores $S_{\text{sta}}(x, E_{k,l})$; while their representations do not collapse or rapidly decay along perturbation directions, leading to large expressiveness scores $S_{\text{exp}}(x, E_{k,l})$. In contrast, brittle shortcuts or filler steps produce large distribution shifts or rapid representation decay, leading to low $R_{\text{cf}}(x, E_{k,l})$ even when the final answer is accidentally correct. Thus, the reward provides a fine-grained signal that separates process validity from outcome correctness and steers the model toward robust, reusable reasoning patterns.

3.3 Policy Optimization

In this subsection, we illustrate the policy optimization of GC^2PO. Specifically, we integrate the proposed causal counterfactual reward with the outcome reward in a GRPO-style objective, considering both reasoning effectiveness and answer correctness. For each candidate y_k and episode index l, we define an episodic score that combines both rewards:

$$S_{k,l}(x) = \frac{1}{L_k} R_{\text{out}}(x, y_k) + \lambda_{\text{cf}} R_{\text{cf}}(x, E_{k,l}), \tag{4}$$

where λ_{cf} is a hyperparameter and $R_{\text{out}}(x, y_k) \in \{0, 1\}$ is the outcome reward that is evenly distributed across episodes. We then distribute episodic scores to tokens. For the t-th token of trajectory k, let l

Table 1: Pass@1 performance on mathematical reasoning datasets. We compare base models trained with different approaches. The best results are highlighted in **bold**. "()" denotes the improvement over models without fine-tuning. More results are provided in **Appendix E**.

Methods	AIME 2024	AIME 2025	AMC 2023	MATH500	MinervaMATH	GSM8K	Average Results
DeepScaleR-1.5B-Preview	42.8	36.7	83.0	85.2	24.6	89.6	60.3
+ GRPO (Shao et al., 2024)	44.5 (+1.7)	39.3 (+2.6)	81.5 (-1.5)	84.9 (-0.3)	24.7 (+0.1)	89.4 (-0.2)	60.7 (+0.4)
+ length penalty (Arora & Zanette, 2025)	40.3 (-2.5)	30.3 (-6.4)	77.3 (-5.7)	83.2 (-2.0)	23.0 (-1.6)	88.5 (-1.1)	57.1 (-3.2)
+ ReST-MCTS (Zhang et al., 2024)	45.5 (+2.7)	39.5 (+2.8)	83.4 (+0.4)	84.8 (-0.4)	23.9 (-0.7)	89.9 (+0.3)	61.2 (+0.9)
+ GVPO (Zhang et al., 2025a)	46.1 (+3.3)	39.7 (+3.0)	83.6 (+0.6)	85.7 (+0.5)	25.3 (+0.7)	90.4 (+0.8)	61.8 (+1.5)
+ Dr.GRPO (Liu et al., 2025b)	45.8 (+3.0)	39.6 (+2.9)	82.1 (-0.9)	85.3 (+0.1)	25.1 (+0.5)	90.0 (+0.4)	61.3 (+1.0)
+ GCPO (Gu et al., 2025)	46.7 (+3.9)	40.3 (+3.6)	84.1 (+1.1)	86.3 (+1.1)	25.9 (+1.4)	90.5 (+0.9)	62.3 (+2.0)
+ MRT (Qu et al., 2025)	47.2 (+4.4)	39.7 (+3.0)	83.1 (+0.1)	85.1 (-0.1)	24.2 (-0.4)	89.9 (+0.3)	61.5 (+1.2)
+ GC^2PO (Ours)	**49.3 (+6.2)**	**40.6 (+3.9)**	**86.1 (+3.1)**	**88.3 (+3.1)**	**27.5 (+2.9)**	**91.6 (+2.0)**	**63.9 (+3.6)**
DeepSeek-R1-Distill-Qwen-1.5B	28.7	26.0	69.9	80.1	19.8	83.4	51.3
+ GRPO (Shao et al., 2024)	29.8 (+1.1)	27.3 (+1.3)	70.5 (+0.6)	80.3 (+0.2)	22.1 (+2.3)	84.5 (+1.1)	52.4 (+1.1)
+ length penalty (Arora & Zanette, 2025)	27.5 (-1.2)	22.6 (-3.4)	64.4 (-5.5)	77.1 (-3.0)	18.8 (-1.0)	82.6 (-0.8)	48.8 (-2.5)
+ ReST-MCTS (Zhang et al., 2024)	30.5 (+1.8)	28.6 (+2.6)	71.1 (+1.2)	80.4 (+0.3)	20.3 (+0.5)	84.8 (+1.4)	52.6 (+1.3)
+ GVPO (Zhang et al., 2025a)	30.6 (+1.9)	28.2 (+2.2)	71.5 (+1.6)	80.5 (+0.4)	23.1 (+3.3)	85.0 (+1.6)	53.2 (+1.8)
+ Dr.GRPO (Liu et al., 2025b)	30.4 (+1.7)	28.4 (+2.4)	71.3 (+1.4)	80.8 (+0.7)	22.9 (+3.1)	85.0 (+1.6)	53.1 (+1.8)
+ GCPO (Gu et al., 2025)	31.0 (+2.3)	29.0 (+3.0)	71.8 (+1.9)	81.6 (+1.5)	23.4 (+3.6)	85.3 (+1.9)	53.7 (+2.4)
+ MRT (Qu et al., 2025)	30.3 (+1.6)	29.3 (+3.3)	72.9 (+3.0)	80.4 (+0.3)	22.5 (+2.7)	84.7 (+1.3)	53.4 (+2.0)
+ GC^2PO (Ours)	**33.8 (+5.1)**	**30.7 (+4.7)**	**74.3 (+4.4)**	**85.3 (+5.2)**	**25.4 (+5.7)**	**86.5 (+3.1)**	**55.9 (+4.6)**
DeepSeek-R1-Distill-Qwen-7B	55.5	50.2	85.1	87.4	42.1	91.6	68.6
+ GRPO (Shao et al., 2024)	56.9 (+1.4)	51.7 (+1.5)	85.5 (+0.4)	87.7 (+0.3)	43.5 (+1.4)	92.1 (+0.5)	69.6 (+0.9)
+ length penalty (Arora & Zanette, 2025)	53.8 (-1.7)	46.9 (-3.3)	81.2 (-3.9)	83.7 (-3.7)	39.5 (-2.6)	91.1 (-0.5)	66.0 (-2.6)
+ ReST-MCTS (Zhang et al., 2024)	57.1 (+1.6)	52.4 (+2.2)	85.7 (+0.6)	87.9 (+0.5)	42.8 (+0.7)	92.0 (+0.4)	69.7 (+1.0)
+ GVPO (Zhang et al., 2025a)	57.5 (+2.0)	52.1 (+1.9)	86.3 (+1.2)	88.5 (+1.1)	44.2 (+2.1)	92.9 (+1.3)	70.3 (+1.6)
+ Dr.GRPO (Liu et al., 2025b)	57.4 (+1.9)	52.3 (+2.1)	86.4 (+1.3)	88.2 (+0.8)	44.0 (+1.9)	92.3 (+0.7)	70.1 (+1.5)
+ GCPO (Gu et al., 2025)	58.3 (+2.8)	53.0 (+2.8)	87.3 (+2.2)	89.1 (+1.7)	45.0 (+2.9)	92.6 (+1.0)	70.9 (+2.2)
+ MRT (Qu et al., 2025)	57.0 (+1.5)	52.4 (+2.2)	86.0 (+0.9)	88.4 (+1.0)	44.3 (+2.2)	92.2 (+0.6)	70.1 (+1.4)
+ GC^2PO (Ours)	**59.1 (+3.6)**	**54.3 (+4.1)**	**88.2 (+3.1)**	**89.8 (+2.4)**	**45.6 (+3.5)**	**93.6 (+2.0)**	**71.8 (+3.1)**

be the episode index such that $y_{k,t} \in E_{k,l}$. We define an unnormalized surprise weight under the old policy, i.e., $\tilde{w}_{k,t}(x) = -\log \pi_{\theta_{\text{old}}}\big(y_{k,t} \mid x, y_{k,<t}\big)$, and normalize it as $w_{k,t}(x)$. The reward is:

$$r_{k,t}(x) = S_{k,l(t)}(x)\, w_{k,t}(x), \quad \text{s.t. } w_{k,t}(x) = \frac{\tilde{w}_{k,t}(x)}{\sum_{t' \in E_{k,l(t)}} \tilde{w}_{k,t'}(x)}. \tag{5}$$

For each trajectory k, we aggregate the token-level rewards into a single scalar by taking a truncated mean over tokens (i.e., discarding a small fraction of extreme values), denoted by $\tilde{r}_k(x) = \text{TruncMean}\big(\{r_{k,t}(x)\}_{t=1}^{T_k}\big)$. Across the K candidates for the same x, we compute the group-normalized advantage $\hat{A}_k(x)$ with group-wise mean $\bar{r}(x)$ and variance $s_r^2(x)$ of these trajectory-level scores. We rescale this advantage back to tokens according to their relative contribution:

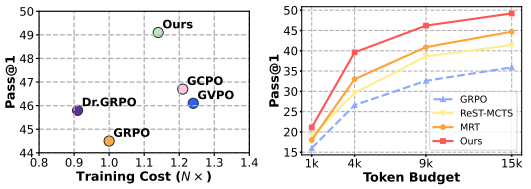
$$\begin{aligned} A_{k,t}(x) &= \hat{A}_k(x)\, \tfrac{r_{k,t}(x)}{\tilde{r}_k(x)}, \text{ s.t. } \hat{A}_k(x) = \tfrac{\tilde{r}_k(x) - \bar{r}(x)}{\sqrt{s_r^2(x)}}, \\ \bar{r}(x) &= \tfrac{1}{K}\textstyle\sum_{k=1}^{K} \tilde{r}_k(x),\ s_r^2(x) = \tfrac{1}{K}\sum_{k=1}^{K}\big(\tilde{r}_k(x) - \bar{r}(x)\big)^2. \end{aligned} \tag{6}$$

In this way, even trajectories with $R_{\text{out}}(x, y_k) = 0$ can receive positive advantages on episodes with high counterfactual rewards, converting previously negative groupwise advantages into useful learning signals for genuinely good but unlucky reasoning. Conversely, for trajectories with $R_{\text{out}}(x, y_k) = 1$, useless episodes obtain relatively small or even negative advantages after normalization, discouraging the model from reinforcing unstable reasoning patterns.

GC^2PO then adopts a GRPO-style optimization objective at the token level:

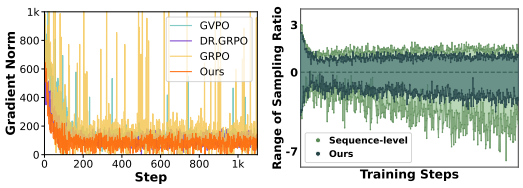
$$\begin{aligned} &\mathcal{J}_{\text{GC}^2\text{PO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \{y_k\}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\Big[\frac{1}{K}\sum_{k=1}^{K}\frac{1}{T_k}\sum_{t=1}^{T_k} g_{k,t}(\theta, x) - \beta_{\text{KL}}\, \mu_{\text{KL}}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big)\Big], \\ &\text{s.t. } g_{k,t}(\theta, x) = \min\Big(\rho_{k,t}(\theta)\, A_{k,t}(x),\ \tilde{\rho}_{k,t}(\theta)\, A_{k,t}(x)\Big), \end{aligned} \tag{7}$$

where β_{KL}, $\rho_{k,t}(\theta)$, and $\tilde{\rho}_{k,t}(\theta)$ are the same as in Eq.2. Based on this, GC^2PO drives the LLMs to generate episodes that are both outcome-successful and counterfactually robust, thereby learning effective and generalizable reasoning strategies across diverse questions and tasks.



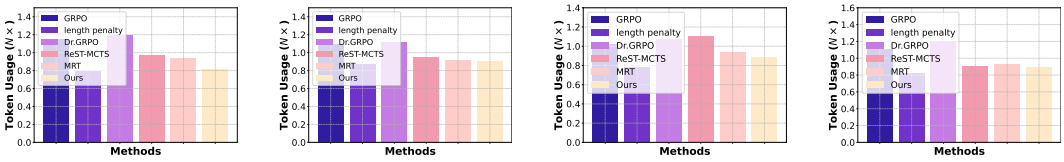
(a) Training time. (b) Token budget.

Figure 4: Trade-off performance analyses.



(a) Gradient norm. (b) Sampling ratio.

Figure 5: Evaluation of training stability.



(a) AIME (b) AMC (c) MATH500 (d) MinervaMATH

Figure 6: Comparison of token usage across different post-training methods.

4 Experiments

4.1 Experimental Settings

We evaluate on a range of reasoning benchmarks, including (i) math, i.e., AIME24-25, AMC, MATH500 (Hendrycks et al., 2021), MinervaMATH (Lewkowycz et al., 2022), GSM8K (Cobbe et al., 2021), GSM-Plus (Li et al., 2024b), (ii) coding, i.e., HumanEval (Chen et al., 2021), (iii) commonsense reasoning, i.e., CommonsenseQA (Talmor et al., 2019), and (iv) formal logic reasoning, i.e., FOLIO (Han et al., 2024); The base models cover different scales, e.g., 1.5B of DeepScaleR-1.5B-Preview and DeepSeek-R1-Distill-Qwen-1.5B, 7B of Qwen2.5-7B-Instruct and DeepSeek-R1-Distill-Qwen-7B, etc. We compare GC^2PO against (i) outcome-reward RL methods, e.g., GRPO (Shao et al., 2024), Dr.GRPO (Liu et al., 2025b), etc., and (ii) process-reward RL methods, e.g., GCPO (Gu et al., 2025), MRT (Qu et al., 2025), etc. Following (Qu et al., 2025; Wang et al., 2025b), DeepScaleR-1.5B-Preview is fine-tuned on 919 AIME problems. DeepSeek-R1-Distill-Qwen-1.5B and 7B are fine-tuned on a random subset of 4,000 NuminaMath QA pairs (Li et al., 2024a). We use a learning rate of 1×10^{-6}, weight decay of 0.01, and batch size 256. The hyperparameters λ_{exp} and λ_{cf} are set to 0.9 and 0.8. All experiments are conducted on A100 GPU clusters.

4.2 Results

Performance and Generalization Analyses. We evaluate GC^2PO across all the benchmarks and base models, measuring both expected pass@1 and total token usage. As shown in **Table 1**, **Figure 6** and **Appendix E.2**, (i) GC^2PO achieves the best performance with fewer tokens, e.g., GC^2PO improves pass@1 by over 2.5% against the baselines; (ii) GC^2PO consistently outperforms baselines on datasets whose distributions differ from the training data, demonstrating great generalization.

Trade-off Performance. We (i) normalize the computational cost of GRPO to $1\times$ and measure total GPU hours across all the methods; and (ii) sample trajectories with a fixed token context window and truncate them at different budgets to evaluate performance. **Figure 4** show that GC^2PO incurs only a modest overhead of $1.2\times$, while achieving the best accuracy with lower budget.

Training Stability. Following common practice (Xiao et al., 2025; Gu et al., 2025), we use the gradient norm as a proxy for policy variance. We also record the sampling ratio of our method compared to previous sequence-level methods. **Figure 5** shows that GC^2PO exhibits more stability, with its gradient norm remaining nearly constant and the sampling ratio achieving no obvious spikes.

Visualization Analysis. To illustrate how GC^2PO changes model behavior, we qualitatively compare models fine-tuned by different methods, visualizing their reasoning traces. **Figure 7** and **Appendix E.5** show that GC^2PO yields more structured reasoning: it decomposes problems into clear subgoals, revisits intermediate steps, and corrects inconsistencies; while GRPO often commits to a single rigid line of reasoning, even when the underlying reasoning is flawed.

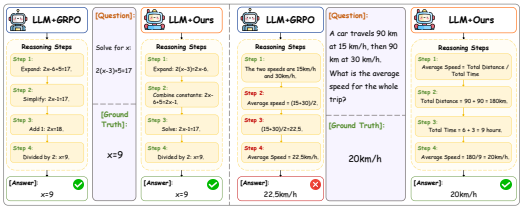
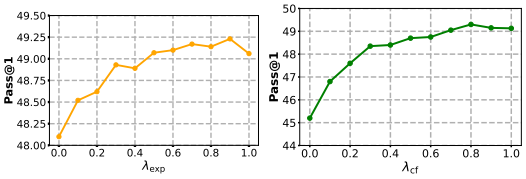


Figure 7: Qualitative Results.



(a) Evaluation of λ_{exp}. (b) Evaluation of λ_{cf}.

Figure 8: Ablation Studies.

Ablation Study. We conduct experiments on the hyperparameters λ_{exp} and λ_{cf}. We search λ_{exp} and λ_{cf} over $[0, 1]$. Note that when they equal 0, we examine two ablated variants: (i) GC^2PO w/o S_{exp}; (ii) GC^2PO w/o S_{nec}; (iii) GC^2PO w/o $S_{\text{sta}}\&S_{\text{exp}}$ (also two terms of $R^{\star}_{\text{cf}}$). **Figure 8** (i) demonstrates the effectiveness of each component within GC^2PO. (ii) shows that the optimal result is at $\lambda_{\text{exp}} = 0.9$ and $\lambda_{\text{cf}} = 0.8$, which are our final configuration. See **Appendix E.4** for more results.

More Results. We also extend our experiments to domains, e.g., code generation, commonsense reasoning, formal logic reasoning, etc., and evaluate more aspects, e.g., the robustness of prompt segmentation and formatting compliance, the plug-and-play nature of GC^2PO, etc. The experimental results in **Appendix E** demonstrate the effectiveness of GC^2PO.

5 Related Work

Complex reasoning remains a challenging capability for LLMs (Jimenez et al., 2024; Snell et al., 2024). To improve reasoning performance, several works (Snell et al., 2024; Wang et al., 2025a;d; Ye et al., 2025b) adopt RL-based post-training methods, which have become a key component underpinning frontier systems such as OpenAI o1 (Jaech et al., 2024) and DeepSeek R1 (Guo et al., 2025). Among these methods, GRPO (Shao et al., 2024) and its variants (Zhao et al., 2025b; Gu et al., 2025; Liu et al., 2025b) have attracted particular interest, as they simplify optimization by using the group's average reward as an advantage baseline, making them well-suited for fine-tuning large-scale models. However, they rely on binary rewards that reflect only outcome correctness, ignoring the structure of the reasoning process (Wang et al., 2025b; Zheng et al., 2025a). To address this issue, some works introduce process rewards to evaluate intermediate steps (Qu et al., 2025; Zeng et al., 2025; Zheng et al., 2025b; Zeng et al., 2025; Wang et al., 2025b; Yuan et al., 2024). However, these methods still primarily target final correctness or rely on outcome labels for training, which may undermine the reasoning reliability and thus affect generalization. To fill this gap, in this paper, we analyze the reasoning of LLMs from a causal perspective. Different from other concurrent works that also leverage causality in LLMs (Zhang et al., 2025e; Vashishtha et al., 2025; Yamin et al., 2025; Lasheras et al., 2025; Chen et al., 2025a; Zheng et al., 2025b), which focus on specific biases (e.g., step length) or stress tests while overlooking the issue of reward misalignment, we focus on the "lucky guess" phenomenon and propose an episodic causal counterfactual reward for generalizable reasoning. Our reward explicitly decouples reasoning quality from answer correctness, encouraging LLMs to acquire generalizable reasoning strategies. See **Appendix D** for more comparisons.

6 Conclusion

In this paper, we explore a critical bottleneck in RL-based post-training: the reward mechanisms of existing methods entangle process validity with final correctness, hindering the acquisition of generalizable reasoning. To overcome this, we establish a causal perspective that formalizes multi-candidate generation as a set of counterfactual experiments and propose Group Causal Counterfactual Policy Optimization (GC^2PO). It introduces a fine-grained reward mechanism to jointly evaluate the robustness and effectiveness of reasoning episodes under mechanism-preserving perturbations. This design decouples reasoning quality from outcome correctness: it provides positive supervision for logically sound reasoning steps, while suppressing spurious shortcuts and lucky guesses. By combining it with outcome reward, we build token-level advantage for optimization, encouraging LLMs to favor reasoning patterns that are both process-valid and answer-correct. Extensive experiments across diverse benchmarks demonstrate the advantages of our method.

AI USAGE STATEMENT

In this work, we used generative AI tools for grammar and language polishing. They were not involved in research ideation, methodology, analysis, or substantive writing. The authors take full responsibility for all contents of the paper.

REFERENCES

Daman Arora and Andrea Zanette. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*, 2025.

Haim Avron. A generalized courant-fischer minimax theorem. 2008.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Thierry Cazenave and Alain Haraux. *An introduction to semilinear evolution equations*, volume 13. Oxford University Press, 1998.

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

Yuefei Chen, Vivek K Singh, Jing Ma, and Ruxiang Tang. Counterbench: A benchmark for counterfactuals reasoning in large language models. *arXiv preprint arXiv:2502.11008*, 2025a.

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025b.

Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

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

George Darmois. Analyse des liaisons de probabilité. In *Proc. Int. Stat. Conferences 1947*, pp. 231, 1951.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep rl: A case study on ppo and trpo. In *International conference on learning representations*, 2019.

Silin Gao, Antoine Bosselut, Samy Bengio, and Emmanuel Abbe. Abstral: Augmenting llms' reasoning by reinforcing abstract thinking. *arXiv preprint arXiv:2506.07751*, 2025.

Ziyin Gu, Jingyao Wang, Ran Zuo, Chuxiong Sun, Zeen Song, Changwen Zheng, and Wenwen Qiang. Group causal policy optimization for post-training large language models. *arXiv preprint arXiv:2508.05428*, 2025.

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.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, et al. Folio: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22017–22031, 2024.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

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. *arXiv preprint arXiv:2103.03874*, 2021.

Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. Followbench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4667–4688, 2024.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues?, 2024. URL https://arxiv.org/abs/2310.06770.

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.

Neeraja Kirtane, Yuvraj Khanna, and Peter Relan. Mathrobust-lv: Evaluation of large language models' robustness to linguistic variations in mathematical reasoning. *arXiv preprint arXiv:2510.06430*, 2025.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

Uriel Lasheras, Elioenai Alves, and Vladia Pinheiro. Interventional and counterfactual causal reasoning for llm-based ai agents: A dataset and evaluation in portuguese. *Procesamiento del Lenguaje Natural*, 74, 2025.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Hai-Liang Li, Akitaka Matsumura, and Guojing Zhang. Optimal decay rate of the compressible navier–stokes–poisson system in. *Archive for Rational Mechanics and Analysis*, 196(2):681–713, 2010.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13:9, 2024a.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2961–2984, 2024b.

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*, 2023.

Minghui Liu, Aadi Palnitkar, Tahseen Rabbani, Hyunwoo Jae, Kyle Rui Sang, Dixi Yao, Shayan Shabihi, Fuheng Zhao, Tian Li, Ce Zhang, et al. Hold onto that thought: Assessing kv cache compression on reasoning. *arXiv preprint arXiv:2512.12008*, 2025a.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025b.

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

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

Robert Ness, Kaushal Paneri, and Olga Vitek. Integrating markov processes with structural causal modeling enables counterfactual inference in complex systems. *Advances in Neural Information Processing Systems*, 32, 2019.

Michael Oberst and David Sontag. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pp. 4881–4890. PMLR, 2019.

Tej Deep Pala, Panshul Sharma, Amir Zadeh, Chuan Li, and Soujanya Poria. Error typing for smarter rewards: Improving process reward models with error-aware hierarchical supervision. *arXiv preprint arXiv:2505.19706*, 2025.

Amnon Pazy. *Semigroups of linear operators and applications to partial differential equations*, volume 44. Springer Science & Business Media, 2012.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Yuxiao Qu, Matthew YR Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. Optimizing test-time compute via meta reinforcement fine-tuning. *arXiv preprint arXiv:2503.07572*, 2025.

Joel W Robbin and Dietmar A Salamon. *Introduction to differential geometry*. Springer Nature, 2022.

Ahmed R. Sadik and Siddhata Govind. Benchmarking llm for code smells detection: Openai gpt-4.0 vs deepseek-v3, 2025. URL https://arxiv.org/abs/2504.16027.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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.

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

Shuaijie She, Junxiao Liu, Yifeng Liu, Jiajun Chen, Xin Huang, and Shujian Huang. R-prm: Reasoning-driven process reward modeling, 2025. URL https://arxiv.org/abs/2503.21295.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. Prmbench: A fine-grained and challenging benchmark for process-level reward models. *arXiv preprint arXiv:2501.03124*, 2025.

Csaba Szepesvári. *Algorithms for reinforcement learning*. Springer nature, 2022.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019.

Hongze Tan, Jianfei Pan, Jinghao Lin, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*, 2025.

Guiyao Tie, Zeli Zhao, Dingjie Song, Fuyang Wei, Rong Zhou, Yurou Dai, Wen Yin, Zhejian Yang, Jiangyue Yan, Yao Su, et al. A survey on post-training of large language models. *arXiv e-prints*, pp. arXiv–2503, 2025.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Aniket Vashishtha, Qirun Dai, Hongyuan Mei, Amit Sharma, Chenhao Tan, and Hao Peng. Executable counterfactuals: Improving llms' causal reasoning through code. *arXiv preprint arXiv:2510.01539*, 2025.

Chaoqi Wang, Zhuokai Zhao, Yibo Jiang, Zhaorun Chen, Chen Zhu, Yuxin Chen, Jiayi Liu, Lizhu Zhang, Xiangjun Fan, Hao Ma, et al. Beyond reward hacking: Causal rewards for large language model alignment. *arXiv preprint arXiv:2501.09620*, 2025a.

Jingyao Wang, Wenwen Qiang, Zeen Song, Changwen Zheng, and Hui Xiong. Learning to think: Information-theoretic reinforcement fine-tuning for llms. *arXiv preprint arXiv:2505.10425*, 2025b.

Ning Wang, Bingkun Yao, Jie Zhou, Yuchen Hu, Xi Wang, Nan Guan, and Zhe Jiang. Insights from verification: Training a verilog generation llm with reinforcement learning with testbench feedback, 2025c. URL https://arxiv.org/abs/2504.15804.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Thoughts are all over the place: On the underthinking of o1-like llms, 2025d. URL https://arxiv.org/abs/2501.18585.

Zixiao Wang, Jingwei Zhang, Wenqian Zhao, Farzan Farnia, and Bei Yu. Moreaupruner: Robust pruning of large language models against weight perturbations. *arXiv preprint arXiv:2406.07017*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Marco A Wiering and Martijn Van Otterlo. Reinforcement learning. *Adaptation, learning, and optimization*, 12(3):729, 2012.

Changyi Xiao, Mengdi Zhang, and Yixin Cao. Bnpo: Beta normalization policy optimization. *arXiv preprint arXiv:2506.02864*, 2025.

Khurram Yamin, Gaurav Ghosal, and Bryan Wilder. Llms struggle to perform counterfactual reasoning with parametric knowledge. *arXiv preprint arXiv:2506.15732*, 2025.

Junjie Yang, Ke Lin, and Xing Yu. Think when you need: Self-adaptive chain-of-thought learning, 2025. URL https://arxiv.org/abs/2504.03234.

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*, 2025a.

Guanghao Ye, Khiem Duc Pham, Xinzhi Zhang, Sivakanth Gopi, Baolin Peng, Beibin Li, Janardhan Kulkarni, and Huseyin A Inan. On the emergence of thinking in llms i: Searching for the right intuition. *arXiv preprint arXiv:2502.06773*, 2025b.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms, 2025. URL https://arxiv.org/abs/2502.03373.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*, 2024.

Thomas Zeng, Shuibai Zhang, Shutong Wu, Christian Classen, Daewon Chae, Ethan Ewer, Minjae Lee, Heeju Kim, Wonjun Kang, Jackson Kunde, et al. Versaprm: Multi-domain process reward model via synthetic reasoning data. *arXiv preprint arXiv:2502.06737*, 2025.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772, 2024.

Kaichen Zhang, Yuzhong Hong, Junwei Bao, Hongfei Jiang, Yang Song, Dingqian Hong, and Hui Xiong. Gvpo: Group variance policy optimization for large language model post-training. *arXiv preprint arXiv:2504.19599*, 2025a.

Nan Zhang, Yusen Zhang, Prasenjit Mitra, and Rui Zhang. When reasoning meets compression: Benchmarking compressed large reasoning models on complex reasoning tasks. *arXiv e-prints*, pp. arXiv–2504, 2025b.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025c.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning, 2025d. URL https://arxiv.org/abs/2501.07301.

Zhuhan Zhang, Min Jiang, Jun Kong, and Jiayi Li. Llm guided counterfactual reasoning for zero-shot knowledge based visual question answering. *Neurocomputing*, pp. 131828, 2025e.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. Genprm: Scaling test-time compute of process reward models via generative reasoning, 2025a. URL https://arxiv.org/abs/2504.00891.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025b.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025a.

Congmin Zheng, Jiachen Zhu, Jianghao Lin, Xinyi Dai, Yong Yu, Weinan Zhang, and Mengyue Yang. Cold: Counterfactually-guided length debiasing for process reward models. *arXiv preprint arXiv:2507.15698*, 2025b.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks, 2025. URL https://arxiv.org/abs/2503.15478.

Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International conference on machine learning*, pp. 12979–12990. PMLR, 2021.

APPENDIX

The appendix is organized as follows:

- **Appendix A** provides the list of notations.
- **Appendix B** provides proofs and analyses of the theorems in the main text.
- **Appendix C** provides the pseudo-code of our method.
- **Appendix D** provides more discussions and further theoretical analysis about our work.
- **Appendix E** provides the full results and additional experiments.

A LIST OF NOTATIONS

We list the definitions of all notations from the main text as follows:

□ **Symbols of Problem Settings and Policies**

- $x \sim \mathcal{D}$: a question sampled from the distribution $\mathcal{D}$.
- π_θ: the current trainable policy.
- $\pi_{\theta_{\text{old}}}$: the old policy.
- π_{ref}: the reference policy.
- K: the number of candidate trajectories sampled for the same question x.

□ **Symbols of Trajectories and Episodes**

- $y_k = (z_k, a_k)$: the k-th candidate trajectory, consisting of the reasoning tokens z_k and the final answer a_k.
- T_k: the number of tokens.
- L_k: the number of episodes into which the k-th trajectory is partitioned.
- $E_{k,l}$: the l-th episode in the k-th trajectory, corresponding to a complete logical step.
- $T_{k,l}$: the number of tokens in episode $E_{k,l}$, satisfying $\sum_{l=1}^{L_k} T_{k,l} = T_k$.

□ **Symbols of Reward Design**

- $u_{k,l}(x)$: the representation of episode $E_{k,l}$ for question x.
- $q(\cdot \mid u_{k,l}(x)) \in \Delta(\mathcal{A})$: the answer distribution induced by the episode representation $u_{k,l}(x)$, where $\mathcal{A}$ is the answer space.
- $\{g_m\}_{m=1}^{M}$: a family of lightweight perturbation operators in the representation space.
- $\tilde{u}_{k,l}^{(m)}(x) = g_m(u_{k,l}(x))$: the perturbed episode representation.
- $R_{\text{out}}(x, y_k) \in \{0, 1\}$: the outcome reward.
- $R_{\text{cf}}^{\star}(x, E_{k,l})$: the episodic causal counterfactual reward.
- $\hat{R}_{\text{cf}}(x, E_{k,l})$: the Monte Carlo estimate of $R_{\text{cf}}^{\star}(x, E_{k,l})$ based on M perturbation samples.
- $\lambda_{cf} \geq 0$: the weight of the counterfactual reward.
- $S_{k,l}(x)$: the episode-level total score that combines the outcome reward and the counterfactual reward.

□ **Symbols of Policy Optimization**

- $\tilde{w}_{k,t}(x)$, $w_{k,t}(x)$:
  - $\tilde{w}_{k,t}(x)$: the token-level surprise weight.
  - $w_{k,t}(x)$: the normalized surprise weight within the corresponding episode.
- $r_{k,t}(x)$: the reward assigned to token t in trajectory k.
- $\tilde{r}_k(x)$: the truncated mean of token rewards for trajectory k, used to construct the group-level advantage.
- $\bar{r}(x)$, $s_r^2(x)$:
  - $\bar{r}(x)$: the mean of $\{\tilde{r}_k(x)\}_{k=1}^{K}$ for question x.

  - $s_r^2(x)$: the variance of $\{\tilde{r}_k(x)\}_{k=1}^K$ for question x.
- $\hat{A}_k(x)$: the group-normalized advantage of the k-th trajectory.
- $A_{k,t}(x)$: the token-level advantage, obtained by rescaling $\hat{A}_k(x)$ according to the contribution of token t.
- $\rho_{k,t}(\theta)$, $\tilde{\rho}_{k,t}(\theta)$:
  - $\rho_{k,t}(\theta)$: the token-level likelihood ratio between π_θ and π_{ref}.
  - $\tilde{\rho}_{k,t}(\theta)$: the clipped version of $\rho_{k,t}(\theta)$ used in the GRPO-style objective.
- $g_{k,t}(\theta, x)$: the token-level term inside the GRPO-style clipped objective.
- $\beta_{\text{KL}} > 0$: the coefficient of the KL regularization term.
- ϵ: the clipping parameter in the policy update.
- $\mathcal{J}_{\text{GC}^2\text{PO}}(\theta)$: the overall optimization objective of GC^2PO.

B Theoretical Analyses and Proofs

B.1 Proofs and Analyses of Theorem 2.1

Proofs. We derive the mapping from a finite-state Markov Decision Process (MDP) to a Structural Causal Model (SCM) using the Gumbel-Max reparameterization framework as established in (Oberst & Sontag, 2019; Ness et al., 2019). The proof relies on two core pillars: the ability to represent any discrete distribution as a deterministic function of Gumbel noise (**Lemma B.1**), and the property of counterfactual stability (**Lemma B.2**).

Lemma B.1 (Gumbel-Max SCMs (Oberst & Sontag, 2019))**.** *Let $g_1, \ldots, g_K$ be i.i.d. variables sampled from the standard Gumbel distribution Gumbel$(0, 1)$, characterized by the Cumulative Distribution Function (CDF) $F(g) = e^{-e^{-g}}$ and the Probability Density Function (PDF) $f(g) = e^{-(g+e^{-g})}$. For any categorical distribution over K states with probabilities $\{p_1, \ldots, p_K\}$, the random variable $X = \arg\max_{k\in\{1,\ldots,K\}}\{\log p_k + g_k\}$ follows the distribution $P(X = k) = p_k$.*

Proof. Let $\mu_k = \log p_k$, we wish to find the probability that a specific index i is the maximum. The event $X = i$ is equivalent to the joint condition:

$$\mu_i + g_i > \mu_j + g_j, \quad \forall j \neq i \implies g_j < g_i + \mu_i - \mu_j, \quad \forall j \neq i \tag{8}$$

Conditioning on a fixed value of g_i, and leveraging the independence of $\{g_j\}_{j\neq i}$:

$$P(X = i|g_i) = \prod_{j\neq i} P(g_j < g_i + \mu_i - \mu_j) = \prod_{j\neq i} F(g_i + \mu_i - \mu_j) \tag{9}$$

$$= \prod_{j\neq i} \exp\left(-e^{-(g_i+\log p_i - \log p_j)}\right) = \exp\left(-\sum_{j\neq i} e^{-g_i}\frac{p_j}{p_i}\right) \tag{10}$$

Now, we integrate over the density of g_i:

$$P(X = i) = \int_{-\infty}^{\infty} f(g_i)P(X = i|g_i)dg_i \tag{11}$$

$$= \int_{-\infty}^{\infty} e^{-g_i}e^{-e^{-g_i}} \exp\left(-e^{-g_i}\sum_{j\neq i}\frac{p_j}{p_i}\right) dg_i \tag{12}$$

$$= \int_{-\infty}^{\infty} e^{-g_i} \exp\left(-e^{-g_i}\left[1 + \sum_{j\neq i}\frac{p_j}{p_i}\right]\right) dg_i \tag{13}$$

$$= \int_{-\infty}^{\infty} e^{-g_i} \exp\left(-e^{-g_i}\frac{\sum_{j=1}^K p_j}{p_i}\right) dg_i \tag{14}$$

Since $\sum_{j=1}^{K} p_j = 1$, the integral simplifies to:

$$P(X = i) = \int_{-\infty}^{\infty} e^{-g_i} \exp\left(-\frac{e^{-g_i}}{p_i}\right) dg_i \tag{15}$$

Perform a change of variables: let $z = e^{-g_i}$, then $dz = -e^{-g_i} dg_i$. As $g_i \to \infty$, $z \to 0$ and $g_i \to -\infty$, $z \to \infty$:

$$P(X = i) = \int_{0}^{\infty} \exp\left(-\frac{z}{p_i}\right) dz = \left[-p_i e^{-z/p_i}\right]_0^{\infty} = p_i \tag{16}$$

This confirms that the Gumbel-Max reparameterization is observationally equivalent to the categorical distribution. □

Lemma B.2 (Counterfactual Stability (Oberst & Sontag, 2019; Ness et al., 2019))**.** *The Gumbel-Max SCM satisfies counterfactual stability: given an observed outcome $Y_I = i$ under intervention I with probabilities p, then for any alternative intervention I' with probabilities p', the counterfactual outcome $Y_{I'} = j$ (for $j \neq i$) is impossible if $\frac{p'_i}{p_i} \geq \frac{p'_j}{p_j}$.*

Proof. By the structural assignment $Y = \arg\max_k\{\log p_k + g_k\}$, observing $Y_I = i$ provides the following evidence regarding the latent Gumbel noise:

$$\log p_i + g_i > \log p_j + g_j \implies g_j - g_i < \log p_i - \log p_j, \quad \forall j \neq i \tag{17}$$

Assume a counterfactual outcome $Y_{I'} = j$ for some $j \neq i$. This necessitates:

$$\log p'_j + g_j > \log p'_i + g_i \implies g_j - g_i > \log p'_i - \log p'_j \tag{18}$$

Combining Eq.equation 17 and Eq.equation 18:

$$\log p'_i - \log p'_j < g_j - g_i < \log p_i - \log p_j \implies \log \frac{p'_i}{p'_j} < \log \frac{p_i}{p_j} \tag{19}$$

By the monotonicity of the logarithm:

$$\frac{p'_i}{p'_j} < \frac{p_i}{p_j} \iff \frac{p'_i}{p_i} < \frac{p'_j}{p_j} \tag{20}$$

Therefore, if $\frac{p'_i}{p_i} \geq \frac{p'_j}{p_j}$, the condition for $Y_{I'} = j$ is violated, ensuring that the original outcome i persists or shifts only to states with higher relative likelihood increases. □

Equipped with these lemmas, we proceed to the proof of the main theorem. This is a specialization of the main results in (Oberst & Sontag, 2019; Ness et al., 2019). For a fixed input x, the policy π_θ induces an MDP. We construct the SCM $\mathcal{M}$ as follows:

- Variables: Define endogenous variables $V = \{s_0, a_0, \ldots, s_T\}$ and exogenous $U = \{\epsilon_{t,s,a}, \nu_{t,s,a,s'}\}_{t,s,a,s'}$.
- Structural Equations: (i) Action: $a_t = f_\pi(s_t, \epsilon_{t,s_t}) = \arg\max_a\{\log \pi_\theta(a|s_t) + \epsilon_{t,s_t,a}\}$; (ii) Transition: $s_{t+1} = f_P(s_t, a_t, \nu_{t,s_t,a_t}) = \arg\max_{s'}\{\log P(s'|s_t, a_t) + \nu_{t,s_t,a_t,s'}\}$

By Lemma B.1, at each step t, the conditional probability $P(a_t|s_t)$ in $\mathcal{M}$ is exactly $\pi_\theta(a_t|s_t)$, and $P(s_{t+1}|s_t, a_t)$ is exactly the transition kernel. Since the exogenous noise is i.i.d. across states and time, the trajectory distribution $P_{\mathcal{M}}(\tau)$ matches the MDP. Then, an intervention on the policy π to π' or transitions P to P' in the MDP corresponds to an intervention on the structural equations in $\mathcal{M}$. Under an intervention $do(\pi')$, the equation for a_t is replaced by $f_{\pi'}(s_t, \epsilon_{t,s_t})$. Lemma B.1 ensures that the marginal distribution of a_t now follows $\pi'(\cdot|s_t)$. The environment's structural equation f_P remains unchanged, as does the realized noise ν. The resulting counterfactual trajectory is uniquely determined by the fixed realization of (s_0, U) and the new mechanisms. Since the marginals at each step match the modified MDP dynamics (Lemma B.1) and the coupling is stable (Lemma B.2), under the coupling of Common Random Numbers (CRN), the counterfactual distribution in $\mathcal{M}$ coincides with the distribution obtained by simulating the modified MDP under identical stochastic conditions (fixed U). Thus, we complete the proofs.

Specialization to LLM Reasoning Paths In the LLM context, for a fixed input x, we define the state $s_{k,t} = \langle x, z_{k,1:t} \rangle$, where z_k is the sequence of reasoning thoughts of the k-th trajectory y_k. The action a_t is the selection of the next thought unit. Formally, the state transition can be expressed as a recursive concatenation:

$$s_{k,t} = s_{k,t-1} \oplus a_{k,t-1}, \quad s_{k,0} = x \tag{21}$$

where $\oplus$ denotes the concatenation operator. Given policy logits $\mathbf{v}_{k,t} = \pi_\theta(s_{k,t})$, the generation is reparameterized as $a_{k,t} = \arg\max_j \{[\mathbf{v}_{k,t}]_j + \epsilon_{k,t,j}\}$. We introduce a family of local perturbations $\mathcal{P} = \{P_\delta\}$ (e.g., phrasing changes in x). A perturbation $\delta \in \mathcal{P}$ transforms the input x into $x_\delta = \delta(x)$. The outcome R_{out} of a reasoning path $y_k = (z_k, a_k)$ under perturbation δ and noise U is:

$$R_{\text{out}}(x_\delta, y_k, U) = \mathcal{G}(s^{CF}_{k,T_k}(\delta, U)) \tag{22}$$

where $\mathcal{G}$ is a grading function. The total success probability under the observed policy is:

$$P(R_{\text{out}} = 1|x) = \sum_{k=1}^{K} P(y_k|x, \pi_\theta) \mathbb{E}_{U|y_k}[R_{\text{out}}(x, y_k, U)]. \tag{23}$$

We evaluate the robustness of a reasoning pattern y_k:

$$\mathcal{S}_{ro}(y_k) = \mathbb{E}_{\delta\sim\mathcal{P}} \mathbb{E}_{U\sim p(\cdot|y_k,x)} \left[\mathbf{1}(R_{\text{out}}(x_\delta, y_k, U) = 1)\right] \tag{24}$$

For a fixed reasoning logic τ, we define the counterfactual trajectory $y^{CF}(\delta, U)$ under perturbation δ by maintaining the exogenous noise U used in the original generation. The deviation $\Delta(\delta, U) = R_{\text{out}}(x_\delta, y^{CF}, U) - R_{\text{out}}(x, y, U)$ measures the causal sensitivity of the logic. This formula captures whether the logical sequence y_k remains valid when the problem's surface form or environmental constraints shift. Under this causal perspective, for a question x, the sampled trajectories $\{y_k\}_{k=1}^K$ can be viewed as heuristic approximations of parallel counterfactual experiments under shared exogenous noise but different reasoning paths.

Why Shared Exogenous Noise. The exogenous noise U in an SCM represents the intrinsic state or stochastic context of the model at the time of generation. Conceptually, counterfactual comparison corresponds to holding U fixed and varying the decision rule; in practice, group sampling provides multiple alternative trajectories that approximate such comparisons. This is formally justified as follows: if two candidates y_i, y_j arise from the same prompt x, their differences are attributable to the divergent interventions $do(a_{i,t})$ vs $do(a_{j,t})$ applied to the shared latent state U. This transforms the candidate set from a simple collection of independent samples into a structured exploration of *what would have happened if* the model had followed a different line of thought under the same initial conditions.

Reasoning Pattern and Incidental Correctness. Our goal is not merely to identify which trajectory happens to succeed, but to ask whether a given reasoning pattern would still solve a slightly perturbed problem. This leads to the concept of episodic causal counterfactual reward $R^\star_{\text{cf}}(x, E_{k,l})$. An incorrect trajectory that demonstrates durable logic (e.g., a path that fails due to a minor typo but solves the core logical puzzle under various perturbations) is counterfactually more valuable than a lucky correct answer that collapses under the slightest change to the input. Thus, the SCM framework allows us to reward process validity and encourage LLMs to learn enduring reasoning strategies rather than brittle answer patterns.

Causal Perspective on Sampling The quote "sampled trajectories can be viewed as parallel counterfactual experiments" encapsulates the transition from frequentist evaluation to causal introspection. By reparameterizing the generative process as $y = f(x, U)$, we decompose the model's behavior into a fixed mechanism f and a random environment U. Sampling K candidates for a question x approximates the counterfactual query: *"Given the same latent context U that led to y_1, would the model still have succeeded if it had branched into y_2 or y_3?"* This perspective is crucial for LLMs because it shifts the focus from answer matching to logical durability. If a reasoning pattern survives perturbations $\mathcal{P}$ across parallel worlds, it indicates that the model has internalized the causal structure of the task.

B.2 Proofs of Theorem 3.1

Proofs. We stress again that our primary interest is not to identify or disentangle all individual (and independent) latent factors, denoted as z_j, but instead to effectively separate the causal content c from

the spurious style factors s, such that the causal variables can be subsequently recovered for robust decision-making.

We first define this distinct notion of block-identifiability.

Definition B.3 (Block-Identifiability)**.** *A representation $u(x)$ block-identifies the causal variables c if there exists an invertible function (diffeomorphism) h such that $u(x) = h(c)$.*

This implies that the representation captures all information about c and nothing else, up to a reversible transformation.

Then, we provide the theoretical setup for this theorem. To establish identifiability, we adopt a latent variable model perspective following (Pearl, 2009; Schölkopf et al., 2021). For a fixed question x, we assume the episode $E_{k,l}$ is generated by underlying latent factors $z \in \mathcal{Z} \subseteq \mathbb{R}^n$. Note that in this subsection, the meaning of z here is different from Subsection 2.1, but in a measure-theoretic sense. We decompose the latent space into two blocks: $z = (c, s)$, where $c \in \mathcal{C}$ denotes the causal factors (invariant content) that determine the true reasoning logic and answer; $s \in \mathcal{S}$ denotes the spurious factors (style or noise) that may vary under perturbations but should not affect the core logic. The representation $u_{k,l}(x)$ extracted from the LLM can thus be viewed as a function of these latent factors, denoted as $u(z) = u(c, s)$. The perturbation operators $\{g_m\}$ acting on the representation space are modeled as inducing transitions in the style space $s \to \tilde{s}$ while preserving c.

Theorem 3.1 establishes that maximizing the episodic causal counterfactual reward $R^\star_{\text{cf}}$ leads to the block-identifiability of the causal factors c. The proof proceeds in three main steps: (i) Characterization of Global Optima (Darmois Construction): We first demonstrate the existence of an ideal representation based on the Darmois construction that achieves the global maximum of the reward. This establishes that any optimal policy must learn a representation that is both almost surely invariant to perturbations (maximizing S_{sta}) and follows a maximum-entropy distribution (maximizing S_{exp}). (ii) Independence from Spurious Factors via Stability: Using a contradiction argument based on differentiability and the support of the perturbation distribution, we prove that the stability objective S_{sta} strictly enforces the representation to be independent of the spurious style variables s. This reduces the representation to a function of causal factors c only, i.e., $u(z) = \psi(c)$. (iii) Block-Identifiability via Expansion: Finally, we utilize the expansion objective S_{exp} to show that the induced mapping $\psi(c)$ must be a diffeomorphism (invertible). By invoking results from measure theory and differential topology regarding entropy-maximizing maps, we conclude that c can be recovered from $u(z)$ up to an invertible transformation.

Step 1. Let the latent variable $z = (c, s) \in \mathcal{Z}$ generate the episode $E_{k,l}$. We aim to find the properties of the mapping $u : \mathcal{Z} \to \mathbb{R}^d$ learned by π^*_θ that maximizes:

$$R^\star_{\text{cf}} = S_{\text{sta}} + \lambda_{\text{exp}} S_{\text{exp}}. \tag{25}$$

Consider the Darmois construction (Hyvärinen & Pajunen, 1999; Darmois, 1951), let $F_i(c_i|c_{1:i-1})$ be the conditional cumulative distribution function (CDF) of the i-th causal factor. We construct a mapping $d : \mathcal{C} \to (0, 1)^{n_c}$ defined by $d_i(c) = F_i(c_i|c_{1:i-1})$. By construction, the random variable $\hat{c}^* = d(c)$ is uniformly distributed on the unit hypercube $(0, 1)^{n_c}$.

We claim that a representation $u^*(z) = d(c)$ (embedded into $\mathbb{R}^d$) attains the global maximum of $R^\star_{\text{cf}}$:

- S_{sta}: Since perturbations g_m only affect the style $s \to \tilde{s}$ while leaving c invariant, and $u^*(z)$ depends only on c, we have $u^*(c, s) = u^*(c, \tilde{s})$. Consequently, $q(\cdot|u^*(z)) = q(\cdot|u^*(\tilde{z}))$, implying the distance norm is exactly 0. Thus, S_{sta} achieves its maximum value of 1.
- S_{exp}: The term S_{exp} acts as a proxy for maximizing the entropy of the representation distribution (analogous to the alignment and uniformity principle). The uniform distribution maximizes differential entropy on a bounded support. Thus, $u^*(z)$ maximizes the expansion term compared to any collapsed or low-dimensional solution.

Therefore, any optimal policy π^*_θ must learn a mapping $u(z)$ that satisfies two conditions almost surely (a.s.) with respect to the data distribution:

$$u(c, s) = u(c, \tilde{s}) \tag{26}$$

and $u(z)$ follows a maximum-entropy distribution.

Step 2. We now prove that the invariance condition derived from maximizing S_{sta} implies that the learned representation $u(z)$ cannot depend on the spurious factors s. Let $u(z) = h(c, s)$ for some smooth function h.

Suppose for the sake of contradiction, that h depends on some style variable s_l. That is, there exists a style index $l \in \{1, \dots, n_s\}$ and a point $z^* = (c^*, s^*) \in \mathcal{Z}$ such that the partial derivative is non-zero:

$$\frac{\partial h}{\partial s_l}(c^*, s^*) \neq 0. \tag{27}$$

Since f and the network are smooth, h is smooth. By continuity of the derivative, there exists a neighborhood (an open ball) around (c^*, s^*) where this derivative remains non-zero. Specifically, there exists $\eta > 0$ such that for $s_l \in (s_l^* - \eta, s_l^* + \eta)$, the function $s_l \mapsto h(c^*, s_{-l}^*, s_l)$ is strictly monotonic (and thus injective).

Define the auxiliary function $\Psi : \mathcal{C} \times \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ as the difference caused by a perturbation:

$$\Psi(c, s, \tilde{s}) := \|h(c, s) - h(c, \tilde{s})\|_2. \tag{28}$$

The strict monotonicity implies that if we pick two distinct style values $s_l, \tilde{s}_l$ from the η-neighborhood, $\Psi > 0$. Specifically, consider the open sets of style values strictly separated within the neighborhood:

$$\mathcal{U}_s = (s_l^*, s_l^* + \eta) \quad \text{and} \quad \mathcal{U}_{\tilde{s}} = (s_l^* - \eta, s_l^*). \tag{29}$$

For any $s \in \mathcal{U}_s$ and $\tilde{s} \in \mathcal{U}_{\tilde{s}}$, we have $h(c^*, s) \neq h(c^*, \tilde{s})$, implying $\Psi(c^*, s, \tilde{s}) > 0$.

Let $\mathcal{O} \subset \mathcal{Z} \times \mathcal{Z}$ be the open set in the joint space corresponding to these neighborhoods. Since Ψ is continuous and positive on $\mathcal{O}$, and the prior density $p(z)$ is positive following (Pearl, 2009; Kusner et al., 2017), we must check if the perturbation distribution allows reaching this region.

By the assumption within the followed (Pearl, 2009; Kusner et al., 2017) of perturbation coverage, for any style s, the distribution of perturbed styles $\tilde{s}$ covers a neighborhood of s. Specifically, the conditional density $p(\tilde{s}|s)$ is strictly positive in an open set around s. This implies that the measure of the set where perturbation occurs such that $s \in \mathcal{U}_s$ and $\tilde{s} \in \mathcal{U}_{\tilde{s}}$ is strictly positive:

$$\mathbb{P}\left(\Psi(c, s, \tilde{s}) > 0\right) \geq \mathbb{P}(\mathcal{O}) > 0. \tag{30}$$

However, this contradicts the global optimality condition from Step 1, which requires $\|u(z) - u(\tilde{z})\| = 0$ almost surely (i.e., the probability of deviation must be zero). The contradiction implies that $\frac{\partial h}{\partial s_l} \neq 0$ is false almost everywhere. Therefore, $h(c, s)$ is constant with respect to s. The representation depends solely on the causal factors:

$$u_{k,l}(x) = \psi(c), \tag{31}$$

where $\psi : \mathcal{C} \to \mathbb{R}^d$ is a smooth mapping.

Step 3. Finally, we show that the map $\psi(c)$ is invertible, thereby establishing block-identifiability.

From Step 1, maximizing S_{exp} drives the distribution of $u = \psi(c)$ towards the maximum-entropy distribution (uniform) derived from the Darmois construction. Let p_c be the density of the causal factors and p_u be the pushforward density of the representation (uniform and thus regular). Then, we invoke Proposition 5 from (Zimmermann et al., 2021):

Proposition B.4. *Let $\mathcal{M}, \mathcal{N}$ be simply connected, oriented smooth manifolds without boundaries, and let $\psi : \mathcal{M} \to \mathcal{N}$ be a differentiable map. If a random variable $c \sim p_c$ on $\mathcal{M}$ has a regular density, and its pushforward $u = \psi(c)$ has a regular density p_u on $\mathcal{N}$, then ψ is a diffeomorphism.*

In our case, $\mathcal{M} = \mathcal{C}$ and $\mathcal{N} = \text{Im}(\psi)$ (i.e., effectively $(0, 1)^{n_c}$ under the Darmois limit). Since the optimal policy forces p_u to be the uniform distribution (which is regular, i.e., $0 < p_u < \infty$), the mapping ψ must be invertible. Thus, there exists a function ψ^{-1} such that $c = \psi^{-1}(u_{k,l}(x))$. This satisfies **Definition B.3**, proving that π_θ^* block-identifies the true causal variables. Thus, we complete our proofs.

B.3 Proofs and Analyses of Theorem 3.2

We provide the derivation for the bounds stated in **Theorem 3.2** from the perspective of manifolds and differential geometry following (Pearl, 2009; Robbin & Salamon, 2022; Chung, 1997), utilizing

the framework of harmonic analysis on manifolds and spectral theory. It contains the following three steps: (i) First, we treat the stability term S_{sta} as a harmonic energy functional. Using Taylor expansion on the manifold, we show that minimizing this energy forces the directional derivatives along the non-causal fibers to vanish. (ii) Second, we analyze the expressiveness term S_{exp} using spectral graph theory (Chung, 1997). We model the local perturbations as operators acting on the tangent space. Maximizing S_{exp} is shown to be equivalent to maximizing the Rayleigh quotient of the local expansion operator. By invoking the Courant-Fischer Min-Max Theorem (Avron, 2008), we prove that this imposes a lower bound on the smallest singular value of the representation's Jacobian, thereby forbidding topological collapse. (iii) Finally, combining invariance and invertibility guarantees block-identifiability of c.

For the first part, we model the non-causal manifold $\mathcal{S}$ locally as a Riemannian manifold. The stability term S_{sta} approximates the functional smoothness of the policy π on $\mathcal{S}$. The proof proceeds in four steps using local linearization and trace analysis.

Let $z = (c, s)$ be a point on the manifold. The perturbation operator g_m induces a local displacement vector $\xi_m \in T_s\mathcal{S}$ in the tangent space of the non-causal manifold. For small perturbations $\|\xi_m\| \ll 1$, we perform a second-order Taylor expansion of the policy π around z:

$$\pi(z + \xi_m) = \pi(z) + \nabla_s \pi(z)\xi_m + \mathcal{O}(\|\xi_m\|^2). \tag{32}$$

Then, the stability loss $\mathcal{L}_{\text{sta}} = 1 - S_{\text{sta}}$ minimizes the expected squared difference. Substituting the Taylor expansion into the loss $\mathcal{L}_{\text{sta}} \approx \mathbb{E}_m \left[\|\pi(z) - \pi(z + \xi_m)\|_2^2\right] \approx \mathbb{E}_m \left[\|\nabla_s \pi(z)\xi_m\|_2^2\right]$. Using the properties of the trace operator ($\mathbf{x}^\top \mathbf{A}\mathbf{x} = \text{Tr}(\mathbf{A}\mathbf{x}\mathbf{x}^\top)$), we rewrite the expectation:

$$\begin{aligned} \mathbb{E}_m \left[\|\nabla_s \pi(z)\xi_m\|_2^2\right] &= \mathbb{E}_m \left[\text{Tr}(\xi_m^\top \nabla_s \pi(z)^\top \nabla_s \pi(z)\xi_m)\right] \\ &= \text{Tr}\left(\nabla_s \pi(z)^\top \nabla_s \pi(z) \mathbb{E}_m[\xi_m \xi_m^\top]\right). \end{aligned} \tag{33}$$

Let $\Sigma_\xi = \mathbb{E}_m[\xi_m \xi_m^\top]$ be the covariance matrix of the perturbations. Since the perturbations span the tangent space $T_s\mathcal{S}$ (Assumption based on (Pearl, 2009)), Σ_ξ is positive definite with smallest eigenvalue $\lambda_{\min}(\Sigma_\xi) > 0$.

Using the trace inequality $\text{Tr}(AB) \geq \lambda_{\min}(B)\text{Tr}(A)$ for positive semi-definite matrices:

$$\mathcal{L}_{\text{sta}} = \text{Tr}\left((\nabla_s \pi)^\top (\nabla_s \pi)\Sigma_\xi\right) \geq \lambda_{\min}(\Sigma_\xi)\|\nabla_s \pi(z)\|_F^2. \tag{34}$$

Given the condition $S_{\text{sta}} \geq 1 - \epsilon \implies \mathcal{L}_{\text{sta}} \leq \epsilon$ (assuming appropriate scaling τ), we obtain the bound:

$$\|\nabla_s \pi(z)\|_F^2 \leq \frac{\epsilon}{\lambda_{\min}(\Sigma_\xi)} = \mathcal{O}(\epsilon). \tag{35}$$

By the Poincaré inequality (Belkin & Niyogi, 2003), a vanishing gradient norm implies the function is constant along the connected components of the non-causal manifold. Thus, we have $\pi(c, s) \to \pi(c)$.

Next, we proceed to the second step of step. While stability enforces invariance to s, we must ensure the representation $u = \psi(c)$ preserves the structure of c.

Lemma B.5 (Spectral Gap)**.** *If $S_{\text{exp}} \geq \gamma$ for sufficiently large γ, then the Jacobian $J_c = \nabla_c \psi$ has a strictly positive smallest singular value $\sigma_{\min}(J_c) \geq \delta > 0$.*

The proof of this lemma relies on spectral analysis of the expansion operator. Consider the expressiveness term which maximizes the local expansion ratio. Let Δc be a random perturbation in the content space with covariance I (whitened). The induced change in representation is $\Delta u \approx J_c \Delta c$, where $J_c = \nabla_c \psi$. The objective function S_{exp} approximates the expected Rayleigh quotient of the operator $A = J_c^\top J_c$:

$$\mathcal{J}_{\text{exp}} \approx \mathbb{E}_{\Delta c}\left[\frac{\|J_c \Delta c\|_2^2}{\|u\|_2^2}\right] \propto \text{Tr}(J_c^\top J_c) = \sum_{i=1}^{d_c} \sigma_i^2(J_c). \tag{36}$$

We need to show that maximizing the sum of singular values implies the smallest singular value is non-zero. According to the Courant-Fischer Min-Max Theorem (Avron, 2008), the smallest eigenvalue is characterized by the minimum Rayleigh quotient:

$$\lambda_{\min}(J_c^\top J_c) = \min_{v \neq 0} \frac{v^\top J_c^\top J_c v}{v^\top v}. \tag{37}$$

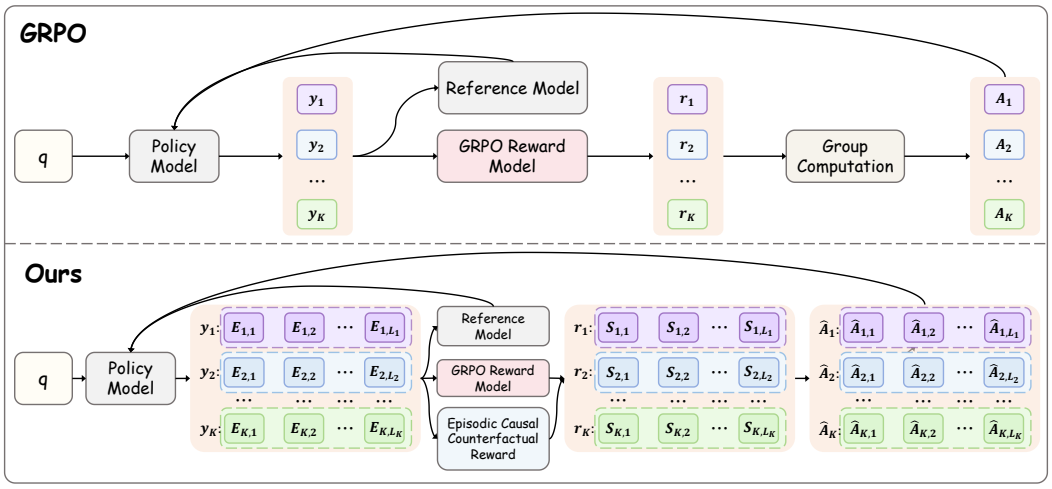


Figure 9: Demonstration of GRPO and our GC^{2}PO. GC^{2}PO proposes a dense episodic causal counterfactual reward instead of the trajectory-level outcome reward to evaluate how well its underlying reasoning pattern generalizes. It then optimizes LLMs by building token-level advantages with it (compare to the trajectory-level advantages of GRPO), ensuring effective and precise reasoning.

We then provide the argument by contradiction. Assume for contradiction that $\sigma_{\min}(J_c) \to 0$. This implies there exists a null direction v_0 in the tangent space of $\mathcal{C}$ such that $J_c v_0 = 0$. The perturbations $\{g_m\}$ in the reward design are stochastic and isotropic. The expectation in S_{exp} integrates over all directions. If the spectrum were degenerate (i.e., $\sigma_{\min} = 0$), the integral of the expansion ratio would be strictly upper-bounded by a value lower than the target γ (assuming γ is set sufficiently high to require expansion in *all* dimensions).

Formally, let the reward impose a constraint that the local volume expansion factor $\det(J_c^\top J_c)^{1/d_c}$ is maximized. By the inequality of arithmetic and geometric means:

$$\frac{1}{d_c}\text{Tr}(J_c^\top J_c) \geq \det(J_c^\top J_c)^{1/d_c} = \left(\prod \sigma_i^2\right)^{1/d_c}. \tag{38}$$

A high reward value implies the geometric mean is non-zero, hence $\sigma_{\min} > 0$. Thus, J_c is full rank, ensuring ψ is a local diffeomorphism.

Combining the above two steps, we establish the convergence to the optimal causal policy. From the first part, we have $\nabla_s \pi^* \approx 0$, implying $\pi^*(c, s) \approx \tilde{\pi}(c)$. From the second part, $\det(\nabla_c u^*) \neq 0$, implying u^* is a lossless encoding of c. Therefore, the learned policy π^* can approximate any function of c.

Specifically, it converges to the oracle policy $\pi_{\text{opt}}(\cdot|c) := \mathbb{E}_s[\pi^*(\cdot|c, s)]$ which is non-causal-marginalized. The KL divergence error is bounded by the variation of the policy along the non-causal manifold. Using the Lipschitz property induced by the gradient bound:

$$\|\pi^*(\cdot|c, s) - \pi_{\text{opt}}(\cdot|c)\| \approx \|\pi^*(\cdot|c, s) - \pi^*(\cdot|c, s_0)\| \leq \int_{s_0}^{s} \|\nabla_s \pi^*\| ds. \tag{39}$$

Since $\|\nabla_s \pi^*\| \leq \mathcal{O}(\sqrt{\epsilon})$ and the manifold is compact with diameter D, the distance is bounded by $D \cdot \mathcal{O}(\sqrt{\epsilon})$. Thus, we have $\mathbb{E}_s[D_{\text{KL}}(\pi_{\text{opt}}(\cdot|c) \,\|\, \pi^*(\cdot|c, s))] \leq \mathcal{O}(\sqrt{\epsilon})$, completing the proofs.

C Pseudo-Code

The pseudo-code of GC^{2}PO is shown in **Algorithm 1**. We also provide an explanation to compare the post-training frameworks of GRPO and GC^{2}PO in **Figure 9**.

Algorithm 1 Pseudo-code of GC^2PO

Require: Initial policy π_θ; prompt distribution $\mathcal{D}$; hyperparameter λ_{cf} and λ_{exp}

for step $= 1, \ldots, N$ **do**
  Sample a mini-batch of questions $D_b \subset \mathcal{D}$
  Set old policy $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
  for each question $x \in D_b$ **do**
    Sample K rollouts $\{y_k\}_{k=1}^{K} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$
    for each rollout $y_k = (z_k, a_k)$ **do**
      Segment z_k into episodes $\{E_{k,l}\}_{l=1}^{L_k}$ via the designed prompt
      Compute outcome reward $R_{\text{out}}(x, y_k)$
      for each episode $E_{k,l}$ **do**
        Extract episode representation $u_{k,l}(x)$ from the LLM
        Sample M perturbations $\{\tilde{u}_{k,l}^{(m)}(x)\}_{m=1}^{M}$ via $\tilde{u}_{k,l}^{(m)}(x) = g_m(u_{k,l}(x))$
        Compute causal counterfactual reward $\hat{R}_{\text{cf}}(x, E_{k,l})$ using Theorem 3.1
        Set episodic score $S_{k,l}(x) = \frac{1}{L_k} R_{\text{out}}(x, y_k) + \lambda_{\text{cf}} \hat{R}_{\text{cf}}(x, E_{k,l})$
      end for
      Amortize $\{S_{k,l}(x)\}_{l=1}^{L_k}$ to token-level rewards $\{r_{k,t}(x)\}_{t=1}^{T_k}$ via Eq.equation 5
      Compute truncated mean $\tilde{r}_k(x) = \text{TruncMean}\left(\{r_{k,t}(x)\}_{t=1}^{T_k}\right)$
    end for
    Compute group-normalized advantages $\{\hat{A}_k(x)\}_{k=1}^{K}$ from $\{\tilde{r}_k(x)\}_{k=1}^{K}$
    Rescale to token-level advantages $A_{k,t}(x) = \hat{A}_k(x) \frac{r_{k,t}(x)}{\tilde{r}_k(x)}$
  end for
  Update π_θ by maximizing $\mathcal{J}_{\text{GC}^2\text{PO}}(\theta)$ via Eq. equation 7
end for

D More Discussion

D.1 More Comparisons

Reasoning of LLMs. Complex reasoning remains one of the most challenging capabilities for LLMs (Jimenez et al., 2024; Zhou et al., 2025; Snell et al., 2024). To improve reasoning performance, many recent works (Snell et al., 2024; Wang et al., 2025a;d; Ye et al., 2025b) adopt RL-based methods, which has become a core component of LLM post-training and underpins frontier systems such as OpenAI o1 (Jaech et al., 2024) and DeepSeek R1, (Guo et al., 2025). Among RL methods, GRPO (Shao et al., 2024) and its variants (Zhao et al., 2025b; Gu et al., 2025; Liu et al., 2025b) are particularly appealing, as they use the group's average reward as a baseline, simplifying optimization and scaling well to large models. However, they rely on binary rewards that capture only outcome correctness and ignore the structure of the reasoning process, limiting what the model can learn (Wang et al., 2025b; Zheng et al., 2025a; Liu et al., 2025b). To mitigate this, several works introduce process rewards that score intermediate steps (Qu et al., 2025; Zeng et al., 2025; Zheng et al., 2025b), but these still focus primarily on trajectory-level correctness and can reinforce solutions that reach the right answer via flawed reasoning, undermining reliability. To address these limitations, we propose a fine-grained causal hierarchy reward for policy optimization that explicitly decouples reasoning quality from answer correctness, encouraging the model to acquire robust reasoning strategies rather than merely imitating final outputs.

Process Reward Models. In contrast to outcome rewards that only assess the final answer, a PRM evaluates the correctness of intermediate reasoning steps (Lightman et al., 2023; Uesato et al., 2022). PRMs can be trained via automated supervision without costly human-annotated step labels (Luo et al., 2024; Setlur et al., 2024; Yuan et al., 2024). Once learned, they can guide test-time search by allocating additional compute to promising trajectories and accelerate RL by scoring the policy's own rollouts (Snell et al., 2024; Muennighoff et al., 2025; Yuan et al., 2024). Recent work (She et al., 2025; Zhao et al., 2025a; Song et al., 2025; Pala et al., 2025; Zhang et al., 2025d) leverages PRMs for RL fine-tuning and reports consistent gains across tasks. However, they still rely heavily on answer correctness: (i) they are typically trained on CoT traces from correct solutions, with limited coverage of erroneous but informative reasoning; and (ii) when scoring, the final correctness is still used as a

primary signal; trajectories with sound reasoning but wrong answers tend to receive low scores, while logically flawed but lucky trajectories may be rewarded, biasing reasoning. In contrast, we propose a fine-grained causal hierarchy reward that focuses on solution strategies rather than only outcomes.

Counterfactual Feedback in LLM Reasoning. In causal inference, a counterfactual describes the outcome that would have occurred for the same underlying system under a different intervention. Some concurrent works explore counterfactual feedback for LLM reasoning (Zhang et al., 2025e; Vashishtha et al., 2025; Yamin et al., 2025; Lasheras et al., 2025; Chen et al., 2025a; Zheng et al., 2025b). For example, (Wang et al., 2025a) introduces counterfactual signals that explicitly adjust for confounding factors in RLHF-style pipelines. CoLD (Zheng et al., 2025b) formulates length bias in PRMs as a causal problem and uses counterfactual interventions on step length to save token cost. VersaPRM (Zeng et al., 2025) and PRMBench (Song et al., 2025) introduce stress tests where counterfactual perturbations expose failure modes. These works collectively highlight the value of counterfactual feedback for LLM reasoning. However, they either operate at the level of benchmark construction or focus on specific bias factors (e.g., length), and ignore the core issue of reasoning we explore in this work, i.e., LLMs are incentivized to imitate final answers rather than to learn stable, transferable reasoning patterns via the existing reward mechanisms. Consequently, trajectories with sound reasoning but incorrect final answers receive little credit, while "lucky guesses" with obvious logical gaps can still be highly rewarded. GC^2PO fills this gap by using counterfactual perturbations in representation space to define fine-grained episodic rewards for reasoning, while retaining the scalability of GRPO-style groupwise optimization.

D.2 Broader Impacts and Limitations

Broader Impacts. This work explores how to make LLM reasoning more accurate and causally reliable across diverse tasks. GC^2PO reframes post-training as an episodic counterfactual credit-assignment problem: instead of rewarding trajectories that merely land on the correct answer, it uses a causal counterfactual reward to favor reasoning steps whose predictions are stable under perturbations and informative across related questions. Extensive theoretical and empirical analyses demonstrate its effectiveness and robustness. This work also opens up exciting new avenues for future research, e.g., the potential to improve robustness and interpretability of LLMs in safety-critical applications (e.g., scientific reasoning, code generation, formal verification), and provides a way to teach LLMs how to think like human rather than merely matching final answers.

Limitations. This work focuses on the various reasoning tasks of LLMs, such as mathematical and code-related tasks, but does not yet explore applications to MLLMs. Our current experiments use mainly the open-source base models at 1.5B, 3B, and 7B scales; larger variants such as the 72B model or $>$100B are not included due to resource and availability constraints. We leave comprehensive studies on multimodal settings and larger base models as important directions for future work.

D.3 Intuition behind the Proposed Reward

To provide a deeper understanding of the proposed episodic causal counterfactual reward, we elaborate on the design intuition behind the stability and expressiveness terms.

What do we intend to reward. The core objective of GC^2PO is to move beyond outcome-based rewards, which often suffer from sparse signals and the shortcut problem (i.e., the model reaching a correct answer via flawed logic). Instead, we aim to reward reasoning episodes that encode invariant and transferable mechanisms. We posit that a high-quality reasoning step should not only lead to the correct answer once but should reflect a robust logical rule that survives counterfactual probes and maintains enough expressive power to generalize to unseen scenarios. Since direct perturbation of input tokens is computationally prohibitive and discrete, we utilize the episode representation $u_{k,l}(x)$ as a proxy for the reasoning mechanism's state and evaluate its generalization in the representation space.

Why introducing the stability term S_{sta}. In causal discovery, a fundamental principle is that a true causal mechanism should be invariant across different environments. We operationalize this through a local counterfactual lens based on the analyses in **Subsection 2.3**: "If the internal state of this reasoning step were slightly different, i.e., simulating a minor variation in context or a different framing of the same logic, would the predictive behavior remain consistent?" By applying a family

of perturbation operators $\{g_m\}$, we create a set of "what-if" scenarios in the representation space. If an episode captures a genuine, mechanism-based rule, the induced answer distribution $q(\cdot|u)$ should be stable across these counterfactual worlds. Conversely, an episode based on brittle shortcuts will exhibit high sensitivity to these probes, indicating that its reasoning is tied to specific noise rather than a stable causal law. In this sense, S_{sta} not only rewards episodes with strong distributional robustness, but also reflects how reliably each episode contributes to correct outcomes when the problem is locally perturbed.

How the stability term S_{sta} is measured. As shown in **Theorem 3.1**, the stability score is defined as:

$$S_{\text{sta}}(x, E_{k,l}) = \mathbb{E}_m\Big[\exp\Big(-\tfrac{1}{\tau}\big|q(\cdot|uk, l(x)) - q^{(m)}(\cdot|x)\big|_2^2\Big)\Big]. \tag{40}$$

The term $\|q - q^{(m)}\|_2^2$ quantifies the behavioral shift. The exponential mapping transforms this shift into a similarity score in $(0, 1]$, where the temperature τ controls the penalty's stringency. A small τ forces the model to learn strictly invariant patterns, while a larger τ allows for more flexibility. It is called stability because only episodes that maintain their predictive integrity in the face of counterfactual perturbations can achieve high stability. This ensures the model assigns higher credit to reasoning steps that are both causally stable and reliably supportive of accurate solutions.

Why introducing the expressiveness term S_{exp}. The expressiveness term proactively enforces that the learned reasoning strategies remain sufficiently rich to transfer across tasks. A reasoning step is only valuable if it carries a strong signal that can withstand the complexity of diverse questions. If the representation u decays too rapidly along perturbation directions, it suggests that the reasoning pattern is structurally fragile and easily obscured by irrelevant nuisance factors (Cazenave & Haraux, 1998; Li et al., 2010). By constraining the decay rate, we drive the LLM to seek reasoning manifolds that are both persistent and informative, ensuring that the strategies learned on one task have the structural depth to be re-applied elsewhere. Meanwhile, optimizing for stability alone carries a risk of representation collapse. A model could achieve perfect stability by producing uninformative, low-entropy distributions (e.g., a uniform distribution) or by driving the representation u into a near-zero norm where perturbations have no effect. Such episodes are stable but useless. To ensure the learned reasoning patterns are sufficiently rich and persistent, we introduce the expressiveness term, which prevents the representation from decaying into a vacuous state.

How the expressiveness term S_{exp} is measured. Drawing from the theory of dynamical systems and semigroups, the optimal decay rate describes how information dissipates over time or space. As shown in **Theorem 3.1**, we define:

$$S_{\text{exp}}(x, E_{k,l}) = \mathbb{E}_m\left[\frac{\big\|\tilde{u}_{k,l}^{(m)}(x)\big\|_2^2}{\big\|u_{k,l}(x)\big\|_2^2 + \varepsilon_u}\right]. \tag{41}$$

By linearizing the perturbation g_m as $\tilde{u}^{(m)} \approx A_m u$, the ratio $\|\tilde{u}^{(m)}\|^2/\|u\|^2$ reflects the energy retention along the direction of u. A high ratio indicates that the representation is persistent and does not rapidly decay toward zero along the perturbation manifold. The constant ε_u prevents numerical instability and discourages the model from shrinking u to exploit the ratio. In other words, if $\tilde{u}^{(m)}$ retains a high proportion of the original norm $\|u\|$, it implies the episode representation is resistant to being pushed into uninformative regions (low-norm zones). A larger S_{exp} thus corresponds to a smaller effective decay rate, rewarding episodes that are structurally robust and capable of sustaining complex logical information across the training distribution. Thus, we call this expressiveness because it ensures the episode maintains a high-capacity representation capable of carrying complex logical information.

Why the combination works. The final reward $R_{\text{cf}}^{\star} = S_{\text{sta}} + \lambda_{\text{exp}} S_{\text{exp}}$ identifies the Pareto optimal reasoning patterns: those that are invariant in their logic (via S_{sta}) and persistent in their representation (via S_{exp}). It ensures the model does not just learn stable but empty or strong but chaotic patterns. Instead, it promotes a form of causal counterfactual consistency, where the model's internal reasoning remains both accurate and rich even when the underlying representation is subjected to the variability of a real-world task distribution.

Why it is called Causal Counterfactual Reward. As mentioned above, the reward is "counterfactual" because it evaluates the model's performance in simulated "what-if" scenarios: What if the internal reasoning state had been slightly different ($\tilde{u}$ instead of u)? It is "causal" because the goal of this

consistency check is to approximate the invariance of the underlying causal reasoning mechanism rather than just fitting surface-level correlations.

D.4 Detailed Illustration of Figure 1

Figure 1 provides a conceptual illustration of the misalignment between outcome correctness and reasoning process quality, which motivates the design of our method. In **Figure 1(a)**, we compare human reasoning with LLMs trained using GRPO and GC^2PO on one training example and two test examples. The training problem has a clear symbolic structure and a fixed derivation path, centered on parameterizing a divisibility relation and optimizing a variable under a linear constraint. Since such problems are common in the training distribution, all three agents are able to produce correct answers. The two test problems introduce substantial changes in surface form and semantic context while requiring the same underlying reasoning skills. The first test problem adopts a physical scenario involving average speed, where correctness hinges on applying the proper definition based on total distance and total time rather than on numerical manipulation. The second test problem is framed as a discrete enumeration task that requires complete coverage of candidate solutions and systematic verification of divisibility constraints. Together, these test cases are designed to evaluate whether a model can transfer core reasoning principles under superficial variations, rather than relying on memorized patterns.

Figure 1(a) provides examples of reasoning trajectories. The human solutions exhibit clear analogical transfer. The parameterization and constraint-checking strategy learned from the training example naturally carries over to the test problems. Humans return to definitions when solving the average speed problem and perform exhaustive enumeration with verification in the constrained counting problem. This indicates that human correctness primarily arises from reusable reasoning structures rather than from surface-level pattern matching. The second column, LLM with GRPO, displays a contrasting behavior. Although it solves the training problem correctly, it often falls back on local heuristics in the test cases. The typical errors include applying the arithmetic mean of speeds in the average speed problem or overlooking constraints during enumeration. These failures are not arbitrary arithmetic mistakes but reflect an over-reliance on surface cues, suggesting that outcome reward encourages strategies that work on the training distribution but do not generalize reliably. In contrast, LLM with GC^2PO demonstrates more stable reasoning transfer across the test problems. Its solutions tend to begin with explicit definitions and constraints, followed by stepwise derivations and validation, which leads to consistent reasoning structures and higher accuracy under surface shifts. This comparison highlights the benefit of process-aware optimization in promoting transferable reasoning strategies.

Figure 1(b) further categorizes model outputs into four representative groups to disentangle outcome correctness from process quality. Near-ideal cases exhibit both correct reasoning and correct answers. Near-miss cases follow a largely sound reasoning process, but incorrect answers may be due to minor output-level deviations. Lucky-guess cases arrive at correct answers despite flawed or incomplete reasoning. Fully-bad cases fail in both process and outcome. This taxonomy makes explicit that outcome-only supervision cannot distinguish high-quality reasoning from accidental success, nor can it properly credit near-miss trajectories with strong underlying logic. More explaination about motivating experiment, i.e., **Figure 1(b)(c)** are provided in **Appendix E.1**.

Taken together, **Figure 1** summarizes the central motivation of this work. Despite strong empirical results, existing methods largely rely on binary outcome rewards and overlook the structure and stability of intermediate reasoning. Even approaches that incorporate process-level signals often remain biased toward final correctness, undervaluing valid reasoning paths that end in failure while rewarding lucky guesses. The phenomena illustrated in **Figure 1** motivate our core question: how to design fine-grained rewards that decouple process validity from final correctness, enabling models to learn stable and transferable reasoning behaviors rather than merely imitating answers.

D.5 Detailed Illustration of Figure 2

Figure 2 provides the SCM regarding the reasoning process of LLMs, and the examples of causal factors and spurious cues. In this subsection, we provide a detailed illustration.

Table 2: Performance comparison on CommonsenseQA and FOLIO datasets.

Model	Method	CommonsenseQA	FOLIO
DeepSeek-R1-Distill-Qwen-1.5B	GRPO	56.4	69.5
	G^2CPO (Ours)	**58.1**	**75.1**
DeepSeek-R1-Distill-Qwen-7B	GRPO	68.2	79.5
	G^2CPO (Ours)	**69.5**	**83.3**

Table 3: Robustness evaluation on the GSM-Plus dataset against adversarial linguistic variations.

Model	Method	GSM8K	GSM-Plus
DeepSeek-R1-Distill-Qwen-1.5B	GRPO	84.5	69.6
	G^2CPO (Ours)	**86.5**	**77.9**
DeepSeek-R1-Distill-Qwen-7B	GRPO	92.1	80.9
	G^2CPO (Ours)	**93.6**	**88.1**

Figure 2(a) formalizes a minimal causal account of why correct answers do not necessarily imply correct reasoning. For a given input question x, the model's output y can be jointly influenced by two latent drivers: invariant causal factors c that encode task-governing mechanisms, and spurious cues s that capture surface patterns correlated with success by chance, e.g., only within the training distribution. Crucially, c and s can be observationally indistinguishable when we only condition on x and verify y, because both may lead to the same correct outcome on seen data. This is why outcome-only supervision tends to blur true correlation with spurious correlation, even when the final answer is correct.

Building on this view, **Figure 2(b)** provides an example. Specifically, the left of **Figure 2(b)** illustrates what we mean by invariant causal structure as a generalizable reasoning mode. The transferable mechanism for average speed is not a particular line in the chain-of-thought, but the stable relational structure $v_{\text{avg}} = (\sum d_i)/(\sum t_i)$ together with correct variable binding and additivity of distance and time. Whether the question provides time then speed, or distance then speed, the same mechanism remains valid and can be instantiated in different algebraic forms, i.e., achieve correct final answers on both seen and unseen questions. Hence, the generalization of LLM reasoning should be attributed to maintaining this underlying mechanism across rephrasings and shifted setups, not to reproducing an identical sequence of intermediate computations. In contrast, the shortcut in the right of **Figure 2(b)** exposes how spurious cues can masquerade as reasoning within the training distribution. Treating "average" as the arithmetic mean of two speeds is a brittle heuristic that happens to coincide with the correct result under specific conditions (e.g., equal time segments), making it observationally equivalent to the true mechanism on the seen question. However, under a shifted condition (e.g., the problem constraints are perturbed, shifting from equal time to equal distance in unseen questions), the heuristic no longer matches the governing physics and fails, yielding an incorrect answer despite following a plausible-looking template.

In post-training, we typically only observe the mentioned seen questions drawn from the training distribution. In these questions, the trajectories driven by c and those triggered by s are often observationally equivalent, i.e., both can reach the same correct answer y and thus receive identical high outcome rewards. Under GRPO, the optimization signal is primarily derived from these result-based rewards, so it cannot reliably distinguish whether a successful trajectory reflects the invariant mechanism ($c \to y$) or merely exploits a brittle shortcut ($s \to y$). Consequently, the policy may be updated toward spurious yet reward-aligned patterns that appear effective on seen data, even though they do not correspond to stable causal factors and will break under distribution shifts. This motivates the need for fine-grained rewards that go beyond final correctness and explicitly favor process-valid, mechanism-consistent reasoning over shortcut-based success.

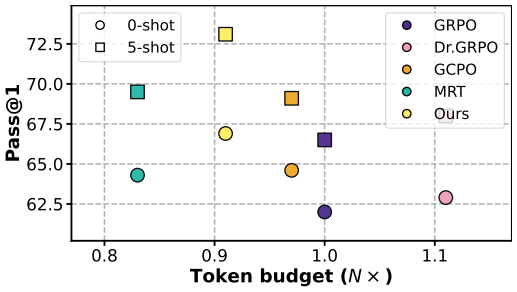


Figure 10: Effectiveness and efficiency analysis of different post-training methods on code reasoning tasks.

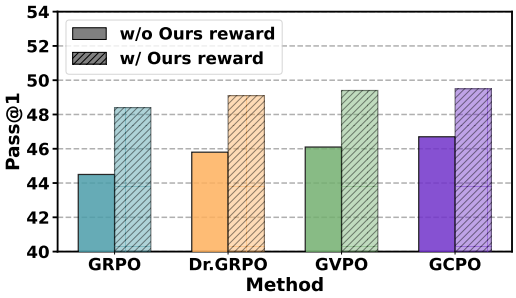


Figure 11: The plug-and-play performance of the episodic causal counterfactual reward on different post-training methods.

Table 4: Effect of combining with MCTS.

Method	AIME	MinervaMATE
GRPO	44.5	24.7
GRPO+MCTS	46.8	25.3
GC^2PO	49.3	26.9
GC^2PO+MCTS	49.5	27.1

Table 5: Applying to inference-only reranking.

Method	MinervaMATH
Best-of-16 + log-likelihood	42.6
Best-of-16 + Qwen2.5-Math-PRM	43.3
Best-of-16 + Skywork-PRM	43.0
Best-of-16 + GC^2PO rerank	44.1

E Additional Experiments and Full Results

E.1 More Details and Analyses of Motivating Experiments

Despite strong empirical performance, many existing reward mechanisms remain tightly coupled to final correctness rather than the intrinsic quality of the reasoning process. To validate this, in **Subsection 2.2**, we conduct a controlled analysis that explicitly disentangles outcome correctness from process validity and examines how different reward mechanisms respond to the two. Here, we will add more details.

We construct an evaluation suite spanning both mathematical reasoning and code generation, drawn from GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), and HumanEval (Chen et al., 2021). Using Qwen2.5-7B-Instruct with specific chain-of-thought prompting, we sample multiple solution trajectories for each problem to expose diverse reasoning attempts under identical inputs. Concretely, for each question we generate $K = 8$ independent trajectories y_k via stochastic decoding (e.g., temperature sampling), and thus obtain a trajectory pool of size $\sum_x K$. For each trajectory, we record two complementary signals. The first is final correctness $f_k \in \{0, 1\}$, determined by an external verifier: for math, we parse the predicted answer and check it against the reference using a symbolic checker; for code, we execute the generated program and evaluate it against unit tests. The second is process validity $p_k \in [0, 1]$, which captures the stepwise soundness of the reasoning rather than the final answer. We estimate p_k through independent annotations of intermediate steps, where human raters are assisted by step-level checkers following (Song et al., 2025; Lightman et al., 2023), e.g., arithmetic verification for math steps and lightweight execution checks for code snippets, so that the validity score reflects whether the intermediate reasoning is coherent, locally correct, and faithful to the stated derivation.

This setup allows us to stratify trajectories by whether they are correct in process and outcome. As illustrated in **Figure 1(b)**, we partition all sampled trajectories into four diagnostic groups: (i) Near-Ideal trajectories exhibit both high process validity and correct final answers; (ii) Near-Miss trajectories follow largely valid reasoning but fail in the final answer (often due to a late arithmetic slip, a wrong case split, or an incorrect terminal step); (iii) Lucky-Guess trajectories reach the correct answer despite weak or inconsistent intermediate reasoning; and (iv) Fully-Bad trajectories are poor in both process and outcome. (ii) and (iii) form a particularly informative contrast: both are partially successful in different dimensions.

We then evaluate two families of reward mechanisms on the same trajectory pool. The first is the outcome reward used in GRPO. The second consists of representative PRMs, including Qwen2.5-

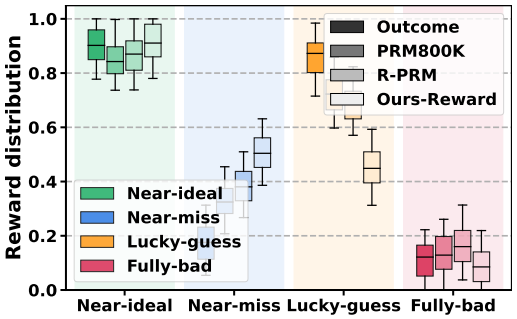


Figure 12: Results of motivating experiments when using GC^2PO.

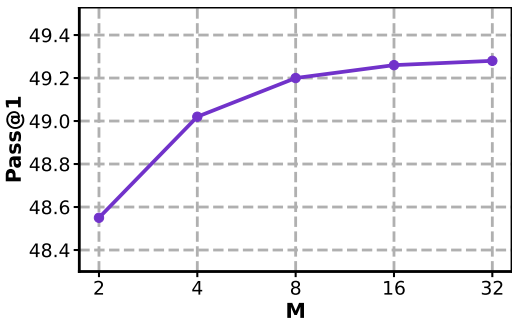


Figure 13: Ablation study of hyperparameter M.

Math-7B-PRM800K and R-PRM-7B, which are designed to score intermediate reasoning quality. We summarize their behavior with two complementary statistics. First, we compute correlations between reward and the two ground signals, i.e., $\mathrm{cor}(R(\cdot), f)$ and $\mathrm{cor}(R(\cdot), p)$, separately on math and code tasks. Second, we compare reward distributions across the four trajectory groups, which reveals whether a reward mechanism can distinguish "good reasoning but wrong answer" from "bad reasoning but correct answer".

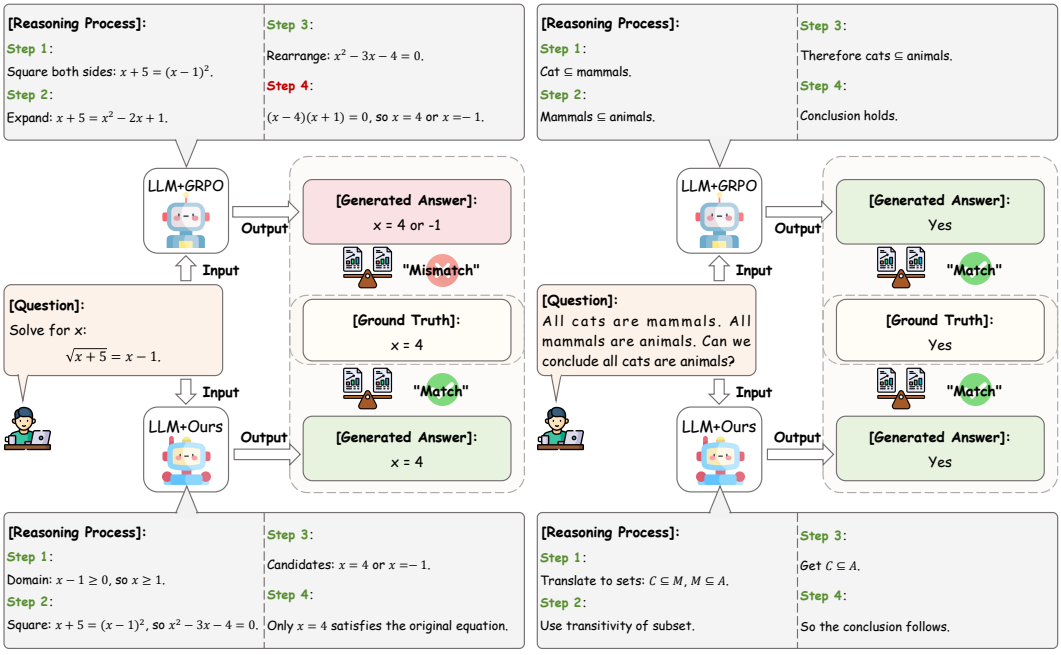
Figure 1(c) presents empirical evidence that existing rewards are still dominated by final correctness. In the correlation analysis (**Figure 1(c) left**), $\mathrm{cor}(R, f)$ is substantially higher than $\mathrm{cor}(R, p)$ for the outcome reward. While PRMs increase alignment with p to some extent, their rewards remain noticeably correlated with f, suggesting that process scoring is still partially entangled with outcome cues. The groupwise reward distributions make this failure mode more concrete (**Figure 1(c) right**). The near-miss trajectories receive rewards close to fully bad trajectories, meaning that trajectories with largely sound reasoning but a late mistake are not accurately credited. Conversely, lucky-guess trajectories obtain rewards comparable to near-ideal solutions, implying that brittle or inconsistent reasoning can be over-rewarded as long as the final answer happens to be correct. Together, these observations highlight a structural limitation of existing reward mechanisms: they do not reliably separate repairable reasoning failures from non-robust shortcut successes, which can hinder the learning of stable and transferable reasoning patterns.

When using the reward of GC^2PO. In contrast, when we replace these sequence-level rewards with our GC^2PO reward, the same trajectory pool becomes much more separable in the intended dimensions (as shown in **Figure 12**). GC^2PO assigns higher credit to Near-Miss trajectories than to Fully-Bad ones, since it explicitly evaluates the robustness and generalizability of the underlying reasoning pattern rather than only the terminal outcome. Meanwhile, GC^2PO suppresses the reward of Lucky-Guess trajectories, because superficially correct answers without coherent intermediate structure are less stable under counterfactual perturbations and therefore receive lower episodic scores. Empirically, this behavior translates into a clearer reward ordering across the four groups and a stronger alignment with process validity, which in turn yields consistent performance gains when GC^2PO is used as the training signal (as demonstrated in **Table 1** and **Figures 4-11**). These results suggest that GC^2PO provides a more faithful supervision signal for learning reusable reasoning patterns, improving both accuracy and transferability beyond what outcome-dominated rewards can offer.

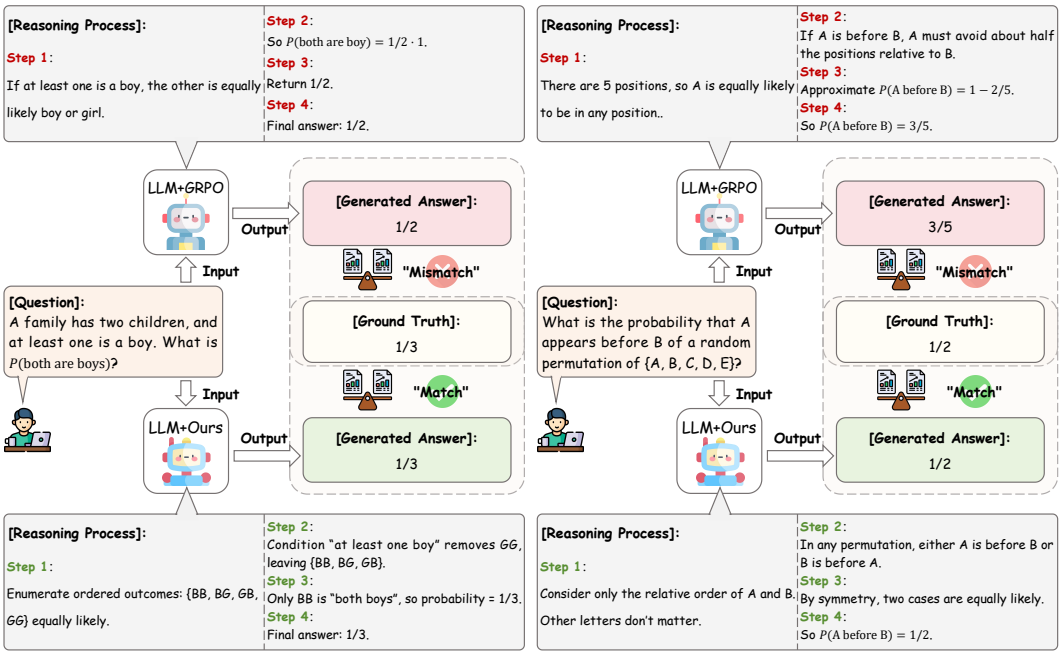
E.2 Performance and Transferability Analyses

To evaluate the effectiveness and generality of the proposed GC^2PO reward, we conduct extensive experiments across diverse reasoning tasks, model scales, and reinforcement-learning variants. In the main text, we focus on mathematical reasoning benchmarks to highlight the core motivation of GC^2PO, i.e., encouraging causally meaningful reasoning trajectories rather than outcome-driven shortcuts. The results in **Table 1**, **Figure 4** demonstrate that GC^2PO consistently improves reasoning accuracy and training stability under comparable or lower test-time compute budgets.

Performance on code reasoning tasks. Beyond mathematical reasoning, we further evaluate GC^2PO on code generation tasks, which require explicit external verification and represent a complementary



(a) Algebra (b) Logic

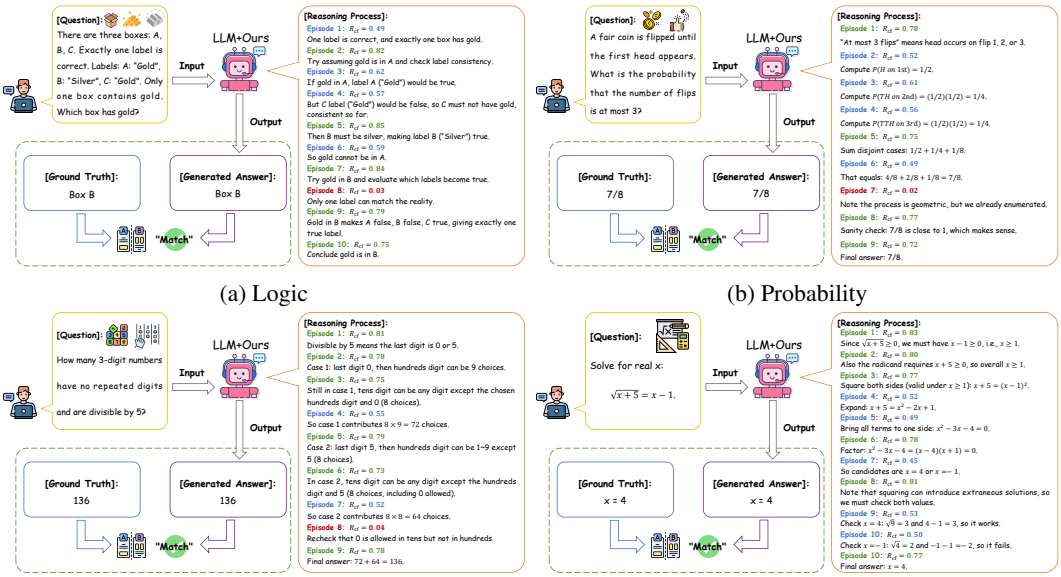


(c) Probability (d) Combinatorics

Figure 14: Examples of qualitative analysis.



Figure 15: Illustration of episode-level reward assignment in GC^{2}PO. Each episode is assessed for its contribution and assigned a graded reward accordingly: green denotes critical steps essential to reaching the correct final answer, blue denotes moderately important steps with limited contribution, and red denotes unimportant or redundant steps that are irrelevant to the final solution.



(a) Logic (b) Probability

(c) Combinatorics (d) Algebra

Figure 16: Illustration of episode-level reward assignment on different types of questions. Each episode is assessed for its contribution and assigned a graded reward accordingly: green denotes critical steps essential to reaching the correct final answer, blue denotes moderately important steps with limited contribution, and red denotes unimportant or redundant steps that are irrelevant to the final solution.

form of structured reasoning. Specifically, we apply Qwen2-7B-Instruct as the base model. We compare the performance of different methods, including GCPO, Dr.GRPO, GCPO, MRT and GC^2PO, on HumanEval using the official execution-based protocol. During evaluation, we generate multiple candidate programs per problem under temperature sampling (temperature $= 0.6$, top-$p = 0.95$), with a strict execution timeout for each attempt. A problem is counted as solved if at least one generated solution passes all unit tests. As shown in **Figure 10**, GC^2PO consistently outperforms outcome-reward-based methods under the same decoding budget, indicating that causally grounded reward signals remain effective even when correctness is determined purely by external execution rather than symbolic matching.

Performance on commonsense reasoning and formal logic reasoning tasks. We extend our experiments to diverse reasoning tasks following the evaluation protocols in existing literature (Liu et al., 2025a; Zhang et al., 2025b). Specifically, we conduct additional experiments on two representative datasets: CommonsenseQA for commonsense reasoning and FOLIO for formal logic reasoning. We employ DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B as our base models. As illustrated in **Table 2**, our method achieves consistent and stable performance gains across both CommonsenseQA and FOLIO compared to the standard GRPO baseline. This confirms that the benefits of our approach are not strictly limited to a single domain, but generalize effectively to various complex reasoning scenarios.

Performance of OOD generalization. The core objective of our proposed framework is to enhance the generalization capabilities of reasoning models. It is important to highlight that our main experiments inherently exhibit cross-distribution characteristics: our models are fine-tuned exclusively on specific subsets (e.g., NuminaMath) but evaluated on independent, distributionally distinct datasets such as MATH500, AMC, and MinervaMATH. The consistent improvements on these unseen datasets already provide strong evidence of OOD generalization. To further stress-test the robustness of our method against adversarial conditions, we evaluate our models on the GSM-Plus. GSM-Plus introduces systematic adversarial distractors and problem restatements specifically designed to evaluate the robustness of mathematical solvers. Recent studies (Gao et al., 2025; Kirtane et al., 2025) have indicated that even state-of-the-art LLMs suffer from significant performance degradation under these linguistic variations. As shown in **Table 3**, while the baseline GRPO experiences substantial

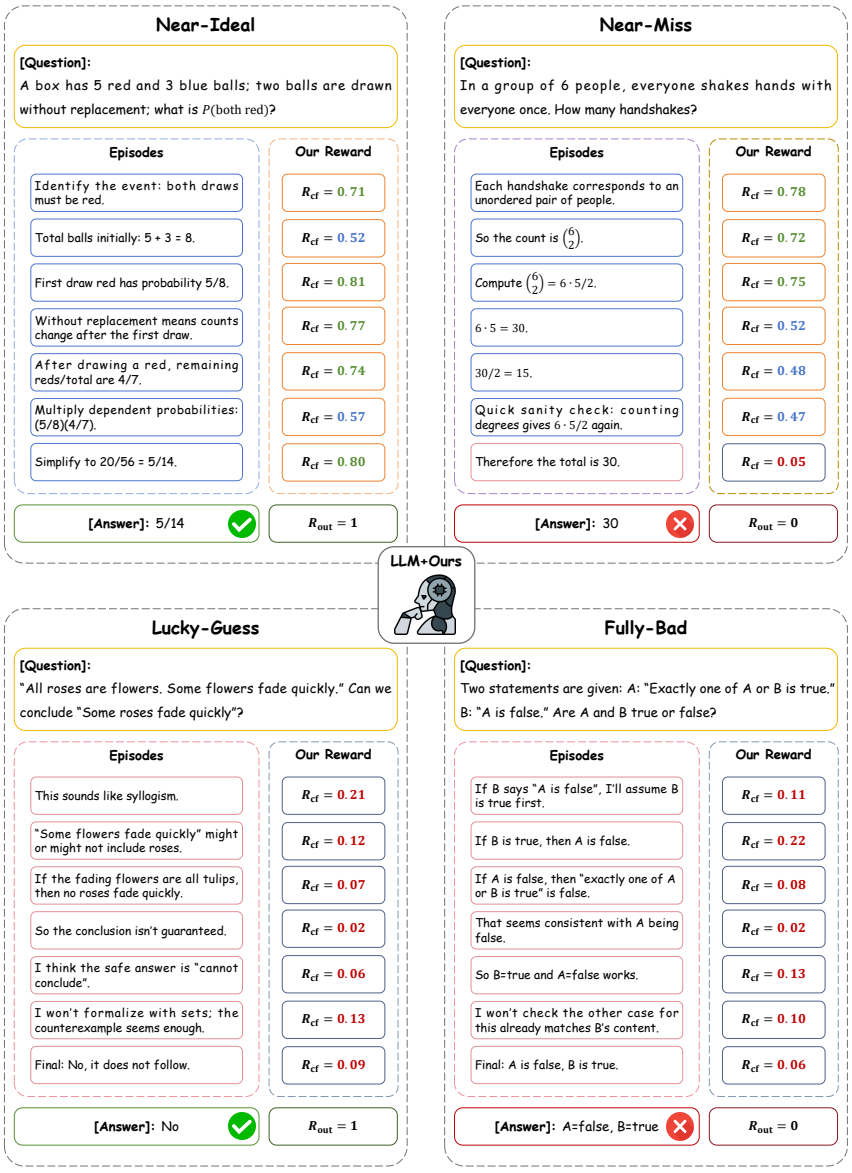


Figure 17: Episode-level reward visualization of GC^{2}PO. Across four representative trajectories (Near-Ideal, Near-Miss, Lucky-Guess, Fully-Bad), GC^{2}PO assigns structured, localized rewards, highlighting episodes that build the correct solution structure while penalizing episodes that fail counterfactual checks, even when the final answer appears plausible or accidentally correct.

performance decay when transitioning from GSM8K to the adversarial GSM-Plus dataset, our method exhibits significantly stronger resilience. The performance drop of our approach is much smaller, confirming that our method not only improves peak accuracy but also substantially enhances the robustness and OOD generalization of the learned reasoning policies.

The plug-and-play nature of GC^2PO. To further assess the plug-and-play nature of GC^2PO, we integrate the proposed reward into several representative GRPO-style variants, including GRPO, Dr.GRPO, GVPO, and GCPO, without modifying their underlying optimization objectives or sampling strategies. In all cases, GC^2PO replaces the original reward component while keeping the remaining algorithmic structure unchanged. As shown in **Figure 11**, across both mathematical and code-related benchmarks, we observe consistent gains in reasoning performance, demonstrating its effectiveness.

Consistent improvement with inference-time reasoning. We additionally analyze whether GC^2PO remains effective when combined with more advanced inference-time reasoning mechanisms following (Wang et al., 2025b). In particular, we apply external search procedures such as MCTS to models fine-tuned with different reward designs and evaluate their performance on challenging benchmarks including AIME and MinervaMATH. The results in **Table 4** show that while outcome-reward-based methods benefit substantially from additional search, GC^2PO-trained models already achieve strong reasoning performance and exhibit only marginal gains from further exploration. This suggests that GC^2PO encourages more self-consistent and transferable reasoning patterns during training, reducing reliance on expensive test-time search. Besides, since GC^2PO proposes a task-agnostic reward over reasoning trajectories, we examine its utility beyond policy optimization. We apply GC^2PO as a reranking signal in inference-only settings, comparing it with likelihood-based reranking and PRM-based reranking under a best-of-K decoding regime. On MinervaMATH, GC^2PO-based reranking yields higher performance than both baselines, as reported in **Table 5**. This result further confirms that GC^2PO captures aspects of reasoning quality that are not reliably reflected by either token-level likelihoods or existing process reward models. These results demonstrate that GC^2PO serves as a general-purpose causal reward that can be seamlessly integrated into different RL fine-tuning frameworks. By explicitly distinguishing between outcome correctness and reasoning validity, GC^2PO enables more stable optimization and more robust reasoning behaviors across tasks, model scales, and post-training paradigms.

E.3 Efficiency and Stability Analyses

Efficiency and stability of post-training. Since the post-training of LLMs is computationally expensive, stable optimization dynamics are crucial for ensuring efficient convergence and avoiding wasted compute. In RL-based post-training, unstable updates can cause excessive policy variance, unreliable convergence behavior, and inefficient use of training samples (Henderson et al., 2018; Engstrom et al., 2019). To quantify training stability, we monitor the gradient norm of policy updates throughout training. Following standard practice in RL-based training, the gradient norm serves as a proxy for the magnitude and variability of policy updates: relatively stable norms indicate well-controlled optimization, whereas large oscillations or spikes often signal unstable or overly aggressive updates (Schulman et al., 2017; Engstrom et al., 2019; Xiao et al., 2025). In addition, we record the sampling ratio of GC^2PO and prior sequence-level methods. This metric reflects how sampling is distributed across candidate trajectories during training; abrupt spikes typically indicate reward-driven mode concentration, where a small subset of trajectories dominates optimization (Schulman et al., 2017; Henderson et al., 2018). As shown in **Figure 5**, GC^2PO exhibits markedly more stable training behavior than prior methods. Meanwhile, the sampling ratio shows almost no pronounced spikes, suggesting that GC^2PO avoids sudden concentration on a few trajectories and maintains a balanced sampling behavior (Henderson et al., 2018). These results demonstrate that GC^2PO leads to smoother optimization dynamics and more reliable convergence, which is especially important given the high computational cost of LLM training.

Stability and robustness of prompt segmentation and formatting compliance. The reward calculation relies on specific formatting tags (e.g., `<episode_i> ... </episode_i>`) to segment the reasoning steps of the base model. As mentioned in **Appendix E.5**, our prompt extends beyond simple step-by-step instructions. We utilize structured templates accompanied by few-shot examples to explicitly enforce formatting constraints (e.g., explicitly stating, *"Each segment must be clearly marked with numbered tags..."*). Extensive prior studies (Jiang et al., 2024; Wei et al., 2023; Guo

et al., 2025) demonstrate that modern LLMs exhibit exceptionally high compliance with explicit format constraints. This is particularly true for the employed DeepSeek-R1-Distill series, which has undergone large-scale format-alignment distillation. To account for extreme exploration scenarios where tag omission (e.g., missing closing tags) might occur, our pipeline incorporates a robust fallback mechanism. If a reasoning trajectory is malformed, the parser does not crash. Instead, it identifies the unclosed text as a single, extended logical unit and proceeds with the holistic counterfactual analysis. Furthermore, as described in **Section 2.1** and **Section 3.3**, our outcome reward R_{out} employs a dual-evaluation metric of "correctness plus formatting". Any trajectory that violates the required format triggers a formatting penalty, actively discouraging the policy from generating malformed outputs. To further ensure the robustness of this mechanism and mitigate the risk of formatting errors during the RL exploration phase, we conduct experiments. We track the formatting error rate of DeepSeek-R1-Distill-Qwen-7B throughout the training process. As shown in Table 6, formatting errors are extremely rare.

Table 6: Formatting error rate of DeepSeek-R1-Distill-Qwen-7B during the RL training phase.

Training Iterations	**Formatting Error Rate**
0 - 50	$< 3.0\%$
50 - 200	$< 0.9\%$
200 - 500	$< 0.1\%$

E.4 Ablation Studies

We conduct several ablation studies to explore how GC^2PO works well, including the effect of each item of GC^2PO, the best parameter configuration, etc. As shown in **Figure 8**, removing any component of GC^2PO consistently degrades performance, verifying the necessity of each term in $R^{\star}_{\text{cf}}$. Moreover, sweeping λ_{exp} and λ_{cf} over $[0, 1]$ reveals a clear optimum at $\lambda_{\text{exp}} = 0.9$ and $\lambda_{\text{cf}} = 0.8$, which we adopt as the default setting.

We further study the sensitivity of GC^2PO to the size of the perturbation family used in the episodic causal counterfactual reward. Recall that GC^2PO relies on a family of lightweight perturbation operators $\{g_m\}_{m=1}^{M}$ in representation space to assess the robustness and generalizability of each reasoning episode. We vary $M \in \{2, 4, 8, 16, 32\}$ while keeping all other hyperparameters fixed. Note that it also determines the computational budget for reward estimation. Specifically, computing $S_{\text{sta}}(x, E_{k,l})$ and $S_{\text{exp}}(x, E_{k,l})$ in **Theorem 3.1** involves expectations over a perturbation distribution, which are generally intractable in closed form. To make the reward computable and scalable, we use Monte Carlo estimators with M perturbation samples.

The results in **Figure 13** show that increasing M improves performance in the low-to-moderate regime. In particular, moving from $M = 2$ to $M = 8$ yields gains in pass@1 accuracy, indicating that a richer perturbation set provides a more reliable estimate of episodic counterfactual stability. Beyond this point, the improvements gradually saturate: $M = 16$ offers only marginal benefits over $M = 8$, and no further gains are observed at $M = 32$, despite the increased computational cost. We also adopt $M = 8$ as the default configuration in experiments.

E.5 Visualization

E.5.1 Qualitative Analysis

To further understand what GC^2PO learns, we conduct a qualitative analysis. Following the same experimental settings in **Subsection 4**, we visualize and compare the reasoning trajectories of different post-training methods across various cases.

The results are shown in **Figure 14**. We can observe that GC^2PO yields trajectories with a cleaner logical structure and higher step efficiency. Its solutions typically follow a small number of semantically complete episodes: the model first identifies the core constraints, then performs the necessary transformations, and finally derives the answer with minimal detours. Each step introduces new information that is required for subsequent inference, and redundant restatements are rare. In contrast,

sequence-level methods may exhibit unfocused exploration. Their trajectories may contain loosely connected statements, speculative intermediate claims, or repeated re-derivations of the same relation. Even when the final answer is correct, the reasoning can include unnecessary loops (e.g., re-checking earlier steps multiple times, rephrasing the same condition in multiple ways, or trying multiple incompatible routes before settling), which increases token cost and makes the logic harder to follow. When the answer is incorrect, these methods are also more likely to sound plausible while drifting away from the true constraint, producing steps that are fluent but not logically useful. These examples suggest that GC^2PO encourages the policy to internalize meaningful step-wise progress.

E.5.2 Reward Assignment

In practice, episode segmentation is determined by semantic completeness and logical separability. Instead of relying on a fixed token length, segmentation is automatically handled by the model during generation and sampling via prompt instructions, resulting in variable-length episodes. Each episode typically corresponds to a coherent reasoning action, such as defining variables, substituting into equations, or performing structural inference. To accommodate tasks with different reasoning granularity, episodes adapt naturally to the problem structure. For questions with long reasoning chains (e.g., mathematical proofs or code generation), a single episode may span multiple tokens to preserve step-level coherence. In contrast, for shorter or more fragmented tasks (e.g., factual queries), the model may compress reasoning into brief logical units, without enforcing unnecessary segmentation. GC^2PO introduces an episodic causal counterfactual reward to assess whether the underlying reasoning pattern of each episode is stable and transferable under mild counterfactual perturbations. We provide qualitative visualizations in **Figure 15** and **Figure 16**, where GC^2PO assigns consistently higher rewards to trajectories composed of semantically coherent and logically robust episodes, while down-weighting near-miss trajectories that contain locally plausible steps but fail under slight variations.

More specifically, GC^2PO produces more structured and localized reward allocation along the reasoning chain. The episodes that establish the key structure of the solution, e.g., correctly setting variables, selecting the governing constraint, or introducing an invariant for subsequent derivations, receive consistently high counterfactual rewards. We provide a set of qualitative analyses as shown in **Figure 17**. In contrast, for near-miss trajectories (correct-looking process but wrong final answer), we find that sequence-level outcome rewards assign uniformly low scores, failing to distinguish partially correct reasoning from truly flawed attempts. PRM-style rewards provide finer granularity but still tend to overweight surface-level fluency: trajectories with locally plausible steps but fragile logical dependencies can obtain relatively high scores. GC^2PO mitigates this issue by penalizing episodes whose validity does not survive counterfactual checks. Typical failure modes include using an implicit assumption that breaks after perturbation, applying a formula outside its conditions, or performing a step that is numerically consistent in the original instance but not structurally justified. We also analyze lucky-guess trajectories, where the final answer is correct but the reasoning contains gaps or invalid steps. GC^2PO assigns noticeably lower rewards to the problematic episodes, preventing such trajectories from dominating optimization. Thus, GC^2PO encourages policies to favor solutions that are not only correct, but also logically robust and reusable, aligning training signals with the desired notion of reasoning quality.

E.5.3 Prompt Configuration for Episode Segmentation

For episode segmentation, we automatically segment the chains by designing specific prompts.

Taking mathematical reasoning tasks as an example, to segment the reasoning chain into fixed episodes (e.g., "segment up to 30"), we add the following instruction to the prompt file:

```
...
<think>
In this section, show your detailed reasoning process.  Break down
    your reasoning into at most 30 logically coherent segments.
```

```
Each segment must be clearly marked with numbered tags in the format <
    episode_1> ... </episode_1>, <episode_2> ... </episode_2>, ..., up
     to <episode_30>.

Ensure each <episode_i> should contain only a single complete logical
    move, such as a definition, a formula derivation, a transformation
    , or a case split.
...
</think>
...
```

For adaptive segmentation (e.g., "break the reasoning chains into K segments"), we add:

```
...
<think>
In this section, show your detailed reasoning process. Break down your
     reasoning into logically coherent segments.

Each segment should be enclosed in <episode_k> ... </episode_k> tags,
    such as:
<episode_1>...</episode_1>
<episode_2>...</episode_2>
...
<episode_K>...</episode_K>

Ensure each  <episode_i> contains only a single complete logical move,
     such as a definition, a formula derivation, a transformation, or
    a case split. Avoid grouping multiple logical steps into one.

...
</think>
...
```