Group Causal Policy Optimization for Post-Training Large Language Models

Ziyin Gu[1,2,*], **Jingyao Wang**[1,2,*], **Ran Zuo**[1,2,3], **Chuxiong Sun**[1,2], **Zeen Song**[1,2], **Changwen Zheng**[1,2], **Wenwen Qiang**[1,2,†]

[1]Institute of Software Chinese Academy of Sciences,
[2]University of Chinese Academy of Sciences,
[3]Communication University of China
{ziyin2020,wangjingyao2023,qiangwenwen}@iscas.ac.cn

Abstract

Recent advances in large language models (LLMs) have broadened their applicability across diverse tasks, yet specialized domains still require targeted post-training. Among existing methods, Group Relative Policy Optimization (GRPO) stands out for its efficiency, leveraging groupwise relative rewards while avoiding costly value function learning. However, GRPO treats candidate responses as independent, overlooking semantic interactions such as complementarity and contradiction. To address this challenge, we first introduce a Structural Causal Model (SCM) that reveals hidden dependencies among candidate responses induced by conditioning on a final integrated output, forming a collider structure. Then, our causal analysis leads to two insights: (1) projecting responses onto a causally-informed subspace improves prediction quality, and (2) this projection yields a better baseline than query-only conditioning. Building on these insights, we propose Group Causal Policy Optimization (GCPO), which integrates causal structure into optimization through two key components: a causally-informed reward adjustment and a novel KL-regularization term that aligns the policy with a causally-projected reference distribution. Comprehensive experimental evaluations on various benchmarks demonstrate that GCPO consistently surpasses existing methods.

Code — https://github.com/ML-TASA/GCPO

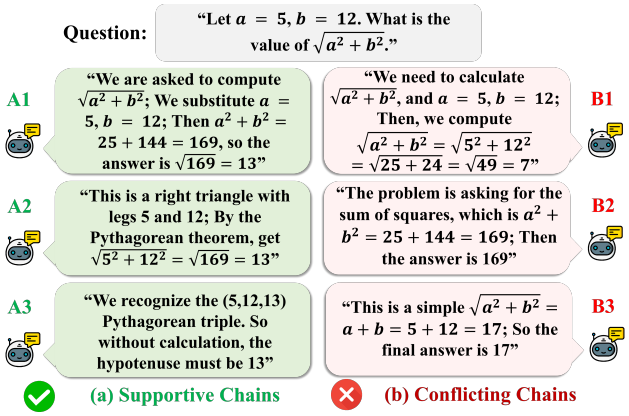


Figure 1: Examples of (a) Supportive chains: A1 provides precise computation, A2 offers geometric insight, and A3 quickly verifies the result via Pythagorean triple recognition; combined, they robustly lead to the optimal answer 13; (b) Conflicting chains: B1 yields 7 due to calculation errors, B2 outputs 169 by omitting the square root, and B3 misinterprets the question to give 17; their conclusions conflict, and mixing them with correct paths introduces errors.

Introduction

Recent advances in large language models (LLMs) have significantly broadened their application potential, demonstrating remarkable capabilities in general tasks (Lai et al. 2025; Zhao et al. 2025; Minaee et al. 2024b; Jaech et al. 2024). However, fully harnessing their practical effectiveness, particularly in specialized domains, requires focused post-training adjustments (Tie et al. 2025). While foundational pre-training establishes linguistic fluency and general reasoning, supplementary methods such as reinforcement learning with human feedback (RLHF) (Bai et al. 2022) are essential for adapting LLMs to specific applications and aligning their outputs with human preferences and ethical norms. Among these approaches, the recently proposed Group Relative Policy Optimization (GRPO) (Shao et al. 2024) has garnered considerable attention due to its significant reduction in computational overhead and memory requirements. By introducing a scalable and efficient training mechanism, GRPO has demonstrated substantial performance gains on many benchmarks (Guo et al. 2025).

While GRPO introduces an efficient mechanism by estimating advantages through groupwise relative rewards, it adopts a simplifying assumption: all candidate responses within a group are treated as independent and unrelated. This design choice helps reduce computational complexity and makes the method more scalable, especially when compared to traditional value-based approaches like PPO (Schulman et al. 2017; Ouyang et al. 2022). However, in many real-world reasoning tasks, responses generated for the same input often contain rich semantic connections. For an example shown in Figure 1, some responses may complement each other by covering different aspects of the problem, jointly

[*]These authors contributed equally.
[†]Corresponding Author.

forming a more complete reasoning chain; others may contradict each other, revealing logical conflicts or alternative interpretations. These interactions—whether supportive or conflicting—are not captured in the current formulation of GRPO. As a result, although GRPO successfully leverages relative reward signals within a group, it may overlook valuable information encoded in the relationships between responses. Incorporating such intra-group dynamics could enable models to better understand the structure of the output space, leading to more nuanced learning and potentially improved alignment with human reasoning preferences.

To address this challenge from a causal perspective, we introduce a Structural Causal Model (SCM) that explicitly captures the relationships between the original query and the generated candidate responses. Specifically, consider a scenario in Figure 2 where a user inputs a query into the LLM, resulting in multiple independently generated candidate answers. Initially, these candidate outputs seem unrelated since each is generated independently based solely on the query. However, if we subsequently use these candidate answers collectively to produce a final, refined response, we unintentionally create a collider structure. In causal inference terms (Pearl, Glymour, and Jewell 2016; Pearl 2009), a collider is a scenario where two or more independent variables influence a common variable, such that conditioning on this common variable makes these previously independent variables become interdependent. In our case, candidate responses are initially independent when conditioned solely on the query. But when these responses jointly influence a final integrated output, conditioning on this final result (the collider) introduces dependencies among the candidate responses. Practically speaking, knowing the content of the final integrated response can reveal previously hidden relationships among candidate answers. For example, one candidate response might provide context missing from another, forming a complementary relationship; another might present contradictory logic, creating a conflicting relationship. Recognizing and explicitly modeling these collider-induced relationships might help the model better leverage hidden structural patterns within generated answers.

Formally, our causal analysis (refer to Section: Causal Analysis and Motivation for more details) provides a rigorous theoretical basis for this intuition. Specifically, Theorem 1 indicates that when the query-response generation process follows a collider structure, predicting an output based on a causally adjusted baseline—that is, the projection of the original predictions onto a subspace that respects this collider structure—will consistently yield improved accuracy. In other words, rather than directly predicting responses based solely on independent evaluations, incorporating a causally informed adjustment significantly enhances prediction performance. Moreover, Corollary 2 complements this by showing that even the original query-based predictions can benefit from incorporating this causally projected baseline. Intuitively, this can be thought of as adding a causal "lens" through which predictions are viewed, enabling the model to correct latent biases or misunderstandings that arise from ignoring structural dependencies.

Motivated by these causal insights, we propose a novel optimization method called Group Causal Policy Optimization (GCPO). Unlike GRPO, which evaluates each candidate response purely based on its reward relative to the group average, GCPO explicitly incorporates causal relationships within the group of generated outputs. Guided by Theorem 1 and Corollary 2, GCPO introduces two major adjustments to the original GRPO framework: (1) a causally-adjusted reward mechanism, and (2) a novel KL-divergence regularization term that explicitly considers causal structures. First, the reward mechanism in GCPO is enhanced by projecting each candidate response onto a causally-informed baseline. Practically, the reward of each candidate answer is adjusted based on how closely it aligns with this causally projected reference. Intuitively, this approach rewards responses that are not only individually strong but also structurally coherent with other responses. Second, to further encourage structural consistency, GCPO introduces an additional KL-divergence regularization term. Specifically, during training, we first compute the model's output distribution conditioned solely on the query. Next, we calculate a causally-adjusted distribution that captures interdependencies among candidate responses. The sum of these two components forms a new reference distribution, representing the model's corrected belief after considering group-level causal structures. By minimizing the KL-divergence between the model's current output and this causally-informed reference, GCPO explicitly guides the model towards structurally consistent predictions. To further illustrate intuitively, the original GRPO method measures divergence by comparing the current policy model to a standard reference model trained without causal adjustments. GCPO, however, measures this divergence against a structurally enhanced baseline, explicitly encouraging the policy model to conform to inferred dependencies among candidate responses. The main contributions of this paper can be summarized as:

- Causal insight into candidate dependencies. We establish that conditioning on a final integrated output induces a collider structure among candidate responses. Theoretically, we prove that projecting predictions onto a causally-informed subspace reduces test error, offering a more reliable baseline than query-only conditioning.
- A causality-aware policy optimization method. We propose GCPO, which enhances GRPO with a causally-adjusted reward and a KL regularizer aligned to a projected reference distribution. This enables structurally consistent and semantically robust policy updates.
- Consistent gains across benchmarks. Experiments on math and code reasoning tasks show that GCPO consistently outperforms GRPO. Ablations confirm the critical role of both proposed components.

Related Work

In recent years, LLMs have made remarkable progress on a wide range of tasks, including question answering (Bottou, Curtis, and Nocedal 2018; Bai et al. 2024; Sun et al. 2024b), code generation (Sadik and Govind 2025; Wang et al. 2025b), and mathematical reasoning (Minaee et al. 2024a; Wang et al. 2025a; Muennighoff et al. 2025; Sun

et al. 2025). However, achieving optimal performance on specialized tasks often requires targeted post-training adaptation (Tie et al. 2025; Sun et al. 2024a, 2021). Common approaches such as Supervised Fine-Tuning (Raffel et al. 2020; Devlin et al. 2019; Zang et al. 2025) and Instruction Tuning (Sanh et al. 2022; Chung et al. 2022; Ouyang et al. 2022) use labeled data or instructional examples to align model outputs with specific objectives, delivering strong results. Nevertheless, these post-training methods are prone to exposure bias and may generalize poorly to novel scenarios (Touvron et al. 2023; Ballon, Algaba, and Ginis 2025).

To address these limitations, reinforcement learning (RL) has been adopted to tailor LLMs for domain-specific applications and align their outputs with human preferences and ethical standards (Ouyang et al. 2022; Bai et al. 2022). Under RL-based strategies, GRPO (Shao et al. 2024) has garnered widespread attention with its efficiency in lowering computational and memory burdens. It introduces a scalable group-wise optimization framework, where policy updates leverage relative advantages within groups of candidate responses. This design enables flexible integration of process rewards and preference signals, resulting in great performance. Building upon GRPO, a number of variants have been proposed, leveraging process-level reward estimation, adaptive reward shaping, and regularization strategies to further improve efficiency and generalization. Specifically, LC-R1 (Cheng et al. 2025) employs a novel combination of a length reward for overall conciseness and a compress reward that is specifically designed to remove the invalid portion of the thinking process. GVPO (Zhang et al. 2025) incorporates the analytical solution to KL-constrained reward maximization directly into its gradient weights, ensuring alignment with the optimal policy. Dr.GRPO (Liu et al. 2025) improves token efficiency while maintaining reasoning performance. L2T (Wang et al. 2025a) proposes an information-theoretic reinforcement fine-tuning framework for LLMs to make the models achieve optimal reasoning with fewer tokens.

However, these exist RL-based policy optimization methods often treat candidate responses as independent, thus ignoring the rich structural and causal relationships that are embedded in the interrelationships among responses. To address this, in this work, we propose GCPO that explicitly models and leverages intra-group dependencies to improve the general coherence and reasoning capability of LLMs.

Causal Analysis and Motivation

This section begins by introducing an SCM. Based on this foundation, we construct a causal analysis framework to evaluate the quality of reasoning strategies in LLMs. We conclude by outlining the motivation that informs the design of the proposed approach.

Causal Analysis

Consider an SCM illustrated in Figure 2. Here, the variable q represents the original input query. The variables $y_0, y_1, \cdots, y_{n-1}$ respectively denote the corresponding outputs obtained by independently feeding the same query q into the function π. The variable y_n is a new output derived by feeding $q, y_0, y_1, \cdots, y_{n-1}$ into the π. Consequently, the SCM includes causal paths: $\{q \rightarrow y_i \rightarrow y_n\}_{i=0}^{n-1}$ and $q \rightarrow y_n$. In addition, the path $q \rightarrow \{y_0, y_1, \cdots, y_{n-1}\}$ forms a fork structure, while the path $\{y_0, y_1, \cdots, y_{n-1}\} \rightarrow y_n$ forms a collider structure (Pearl 2009). These structures lead to two conditional independence relations (Pearl, Glymour, and Jewell 2016): conditioned on q, the variables in $\{y_i\}_{i=0}^{n-1}$ are mutually independent; however, conditioned additionally on y_n, these variables become mutually dependent.

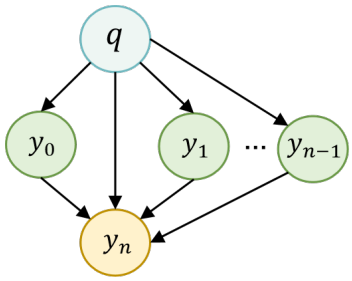


Figure 2: The SCM under our setting. q is the input query, $\{y_0, \cdots, y_{n-1}\}$ represents the set of outputs obtained by feeding q into a LLM n times, and y_n denotes the final output produced by inputting $\{q, y_0, \cdots, y_{n-1}\}$ into a LLM.

From a Bayesian perspective, when a model is trained to optimality by minimizing the cross-entropy loss or mean squared error loss, it can be viewed as estimating the conditional expectation of the output distribution given the input context (Zhang and Bowman 2018; Goodfellow, Bengio, and Courville 2016; Bengio et al. 2003). More precisely, if the function π is trained using cross-entropy loss and achieves optimality, then statistical decision theory implies that $\pi(X) = \mathbb{E}[Y \mid X]$, where X refers to the input in a general sense, and Y refers to the corresponding output. For clarity of distinction, let π denote a function that outputs a probability, similar to the formulation used during training based on cross-entropy loss minimization, and let π^* denote its Bayesian optimal counterpart. Then, we have $\pi(y_0 \mid q) = p_\pi(y_0 \mid q)$ and $\pi^*(q) = \mathbb{E}[y_0 \mid q]$. Based on the conditional independence relations discussed in the previous paragraph, the following conclusion can be drawn:

$$\begin{aligned} &\mathbb{E}\left[\pi^*(x) - \pi^*(q) \mid q, y_{1:n-1}\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[y_0 \mid x\right] - \mathbb{E}\left[y_0 \mid q\right] | q, y_{1:n-1}\right] \\ &= \mathbb{E}\left[y_0 \mid q, y_{1:n-1}\right] - \mathbb{E}\left[y_0 \mid q, y_{1:n-1}\right] = 0, \end{aligned} \tag{1}$$

where $x = \{q, y_1, \cdots, y_n\}$, $y_{1:n-1} = \{y_1, \cdots, y_{n-1}\}$. It is important to note that the variables $y_0, y_1, \cdots, y_{n-1}$ are used as generic placeholders. In other words, Equation (1) still holds when y_0 is exchanged with any $y_i \in \{y_1, \cdots, y_{n-1}\}$. All subsequent results in this section follow this property, and we will not reiterate it in the follows.

Let $\mathcal{F}$ denote the space of square-integrable functions, we can obtain that $\pi^* \in \mathcal{F}$. Let Φ be a functional operator acting on $\pi^*(x)$, defined as the following:

$$\Phi \cdot \pi^*(x) = \mathbb{E}[\pi^*(x) | q, y_1, \cdots, y_{n-1}]. \tag{2}$$

Based on this, we define a causal-related mapping Ψ as: $\Psi = \mathrm{Id} - \Phi$, where Id denotes the identity mapping. Noting that

the output of Φ can be viewed as the image space, while the output of Ψ corresponds to the kernel space associated with Φ. Meanwhile, let X be the random variable of the query and Y be the random variable of the answer, assume $X \times Y \sim p(X, Y)$ where $p(X, Y)$ is the joint probability distribution of X and Y. Given $\pi_1^*, \pi_2^* \in \mathcal{F}$, Δ is defined as:

$$\Delta(\pi_1^*, \pi_2^*) = \mathop{\mathbb{E}}_{p(X,Y)} [Y - \pi_1^*(X)] - \mathop{\mathbb{E}}_{p(X,Y)} [Y - \pi_2^*(X)] . \quad (3)$$

Equation (3) can be interpreted as the test error or the expected risk. Then, the following conclusion can be drawn:

Theorem 1 *Given the condition of Equation (1) and the SCM shown in Figure 2, for $\forall \pi^* \in \mathcal{F}$, the following holds:*

$$\Delta(\pi^*(x), \Psi \cdot \pi^*(x) + \pi^*(q)) \geq 0. \quad (4)$$

The proof of Theorem 1 is presented in the Appendix. We provide an intuitive understanding of Theorem 1. First, the collider structure makes us realize that, although some variables may appear independent on the surface, they could potentially be dependent through a common influence. If this relationship is not captured, it may affect the accuracy of the LLM. Second, $\pi^*(x)$ is tasked with predicting an outcome based on the input. From a causal perspective, $\pi^*(x)$ serves as a generative function. If we know that the data generation process follows a collider structure, we can project the hypothesis space formed by all $\pi^*(x)$ onto a subspace formed by those $\pi^*(x)$ that can recognize the collider structure. This is akin to adding a pair of "glasses" to $\pi^*(x)$, helping it identify latent dependencies that are not immediately apparent, thereby improving its predictive accuracy on new data. Furthermore, $\pi^*(q)$ represents the model's initial prediction in the absence of the collider structure's influence, and it can be viewed as a preliminary estimate of the input. By incorporating this initial estimate, we can further optimize the model, ensuring that the final output does not merely rely on the preliminary estimate but fully considers the inherent structure of the data. Similarly, the follows can also be drawn:

Corollary 2 *Given the condition of Equation (1) and the SCM shown in Figure 2, for π^* and Ψ, the following holds:*

$$\Delta(\pi^*(q), \Psi \cdot \pi^*(x) + \pi^*(q)) \geq 0. \quad (5)$$

The proof of Corollary 2 is provided in the Appendix. Since the intuitive interpretation of Corollary 2 closely parallels that of Theorem 1, we omit a redundant explanation. Together, Theorem 1 and Corollary 2 suggest that when the query generation process involves a collider structure, it is possible to project the hypothesis space of an LLM onto a subspace that better aligns with this structure. By incorporating a baseline function, the model can be further optimized. This approach leverages conditional independence relations encoded in the causal graph, thereby improving the generalization capability of the LLM and enabling more stable and reliable performance on unseen queries.

Motivation Analysis

GRPO has been widely adopted for post-training LLMs due to its efficiency and simplicity. It treats the model as a policy and optimizes it by comparing relative rewards among candidate responses generated for the same query. However, GRPO assumes that all candidates are independent, overlooking potential semantic interactions such as complementarity or contradiction (see Figure 1). This limits the reward signal expressiveness and may hinder LLMs generalization.

From the above causal analysis, while candidate responses are independently sampled from the query, they often influence a final integrated output, thus forming a collider structure. Under this structure, responses become conditionally dependent when the final output is observed. Our theoretical findings (Theorem 1 and Corollary 2) show that projecting predictions onto a causally-informed subspace, expressed as $\Psi \cdot \pi^*(x) + \pi^*(q)$, indeed leads to consistently lower test error than using $\pi^*(q)$ or $\pi^*(x)$ alone.

This insight motivates a principled revision of GRPO's preference mechanism. Instead of favoring candidates purely based on relative rewards, we can additionally consider their alignment with the causally projected output. This adjustment allows the model to exploit latent dependencies among responses, encouraging structurally coherent and semantically accurate outputs. Furthermore, we can introduce a causal regularization term that aligns the policy with a causally-informed reference distribution. Together, these changes form the basis of the following proposed GCPO, a causality-aware optimization framework that enhances model performance by integrating structural reasoning signals into the learning process.

The Proposed Method

In this section, we propose GCPO, a new post-training algorithm for LLMs. GCPO can be viewed as a variant of GRPO, with the primary differences lying in two aspects: the relative advantage function and the KL Divergence.

A Brief Introduction to GRPO

For a given query q, GRPO samples a set of outputs $\{y_0, y_1, \cdots, y_{n-1}\}$ from the old policy $\pi_{\theta_{\text{old}}}$. It then updates the policy π_θ by maximizing the objective:

$$\mathcal{J}_{\text{GRPO}} = \mathbb{E}_{[q\sim P, \{y_0, y_1, \cdots, y_{n-1}\}]} \frac{1}{n} \sum_{i=0}^{n-1} \frac{1}{T_i} \sum_{j=0}^{T_i} \{[\min(R_{i,j}(\theta)A_i, \Xi_{ij} \cdot A_i)] - \beta D_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}})\}, \quad (6)$$

where ϵ and β are hyperparameters, $\Xi_{ij} = \text{clip}(R_{i,j}(\theta), 1-\epsilon, 1+\epsilon)$, $\text{clip}(\cdot)$ is a truncation function ensuring stable updates, P denotes the distribution of queries. Because an LLM generates a output $y_i = (y_{i,1}, \ldots, y_{i,T_i})$ token-by-token in an autoregressive manner, where T_i denotes the token length of y_i, thus, $R_{i,j}(\theta)$ and $D_{\text{KL}}(\cdot)$ are also calculated in a token-by-token manner. Then, the KL divergence term $D_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}})$ is computed as:

$$\frac{\pi_{\text{ref}}(y_{i,j}|q, y_{i,<j})}{\pi_\theta(y_{i,j}|q, y_{i,<j})} - \log \frac{\pi_{\text{ref}}(y_{i,j}|q, y_{i,<j})}{\pi_\theta(y_{i,j}|q, y_{i,<j})} - 1, \quad (7)$$

where π_{ref} is a fixed reference policy and often set to $\pi_{\theta_{\text{old}}}$. The relative advantage A_i is calculated within each sampled group to capture the comparative quality of outputs:

$$A_i = [r_i - \text{mean}(r_0, \cdots, r_{n-1})]/\text{std}(r_0, \cdots, r_{n-1}), \quad (8)$$

where $r_i = \text{reward}(y_i)$ combines task-specific accuracy and formatting rewards, while $\text{mean}(\cdot)$ and $\text{std}(\cdot)$ are the mean and standard deviation over the reward group. At last, the importance ratio $R_{i,j}(\theta)$ is defined as:

$$\pi_\theta(y_{i,j}|q, y_{i,<j})/\pi_{\theta_{\text{old}}}(y_{i,j}|q, y_{i,<j}). \tag{9}$$

Details of the Proposed GCPO

For a query q, GCPO also samples a group of outputs $\{y_0, y_1, \cdots, y_{n-1}\}$ from the old policy $\pi_{\theta_{\text{old}}}$. Different from GRPO, GCPO then input q and $\{y_0, y_1, \cdots, y_{n-1}\}$ into the old policy $\pi_{\theta_{\text{old}}}$ for n times to obtain a final outputs y_n and $\{y_{n,i}\}_{i=1}^{n-1}$. GCPO optimizes the policy model π_θ by maximizing the following objective:

$$\begin{aligned}&\mathcal{J}_{\text{GCPO}} = \mathbb{E}_{[q\sim P,\{y_0,y_1,\cdots,y_n\},\{y_{n,i}\}_{i=1}^{n-1}]}\\&\frac{1}{n}\sum_{i=0}^{n-1}\frac{1}{T_i}\sum_{j=0}^{T_i}\{[\min(R_{i,j}(\theta)B_i, \Xi_{ij}\cdot B_i)]\\&-\beta D_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}})\} - \kappa D_{\text{KL}}(\pi_\theta \parallel \pi'_{\text{ref}}),\end{aligned} \tag{10}$$

where κ is a hyper-parameter, B_i is the newly proposed relative advantage function, π'_{ref} is the newly proposed predefined policy model, and $D_{\text{KL}}(\pi_\theta \parallel \pi'_{\text{ref}})$ is the newly proposed regularizer. In the following, we conduct an in-depth study of the terms B_i and $D_{\text{KL}}(\pi_\theta \parallel \pi'_{\text{ref}})$.

Design of Relative Advantage Function. Formally, we define $B_i = A_i \cdot \Upsilon_i$, where A_i is computed in the same way as in GRPO. We next describe the procedure for designing and computing Υ_i. The design of Υ_i is inspired by Corollary 2. According to Equation (2), when we focus on the answer variable y_0, the expected risk of the output from $\pi^*(q)$ is higher than that of $\Psi\cdot\pi^*(x)+\pi^*(q)$. This suggests, conservatively, that the output quality of $\Psi \cdot \pi^*(x) + \pi^*(q)$ is better than that of $\pi^*(q)$. Based on this observation, the advantage value for each candidate answer corresponding to a given query can be designed as follows: the closer the candidate is to the output of $\Psi \cdot \pi^*(x) + \pi^*(q)$, the higher the advantage value it receives.

The proposed approach faces two practical challenges during implementation: (1) how to approximate $\Psi \cdot \pi^*(x) + \pi^*(q)$; and (2) how to measure the similarity between model outputs. Since π^* represents concrete generated content, which typically includes both intermediate reasoning steps and the final answer, we propose to approximate $\Psi\cdot\pi^*(x)+\pi^*(q)$ using the feature representation of the output. Then, we measure similarity based on the cosine distance between these feature representations. The detailed procedure is as follows: **Step 1:** Approximating $\pi^*_\theta(q)$. Given a answer y_i, we define o_i as the combination of y_i and the intermediate reasoning steps leading to it. We then feed o_i back into π_θ, and extract the hidden representation of the final token from the last layer as the feature representation of o_i, denoted by z_i. Then, let $\bar{z} = \text{mean}(z_0, \cdots, z_{n-1})$, which can be regarded as a Monte Carlo approximation of the output representation of $\pi^*_\theta(q)$; **Step 2:** Approximating $\pi^*_\theta(x)$ for $y_i \in \{y_i\}_{i=0}^{n-1}$. Because that $\pi^*_\theta(x) = \mathbb{E}[y_0 \mid x]$, when we focus on y_i, it equals to that y_0 is exchanged with y_i, and the condition x is exchanged with $x_i = \{q, y_0, \cdots, y_n\} \setminus \{y_i\}$. We then feed x_i into π_θ for n times to obtain the corresponding outputs $\{O_{i,j}\}_{j=1}^n$ and representations $\{Z_{i,j}\}_{j=1}^n$. Let $\bar{Z}_i = \text{mean}(Z_{i,1}, \cdots, Z_{i,n})$, which can be regarded as a Monte Carlo approximation of the output representation of $\pi^*_\theta(x)$ for y_i; **Step 3:** Approximating $\Phi \cdot \pi^*(x)$ for $y_i \in \{y_i\}_{i=0}^{n-1}$. According to the definition of Φ, we first define $y_{n,0} = y_n$ and $x_{i,j} = \{q, y_0, \cdots, y_{n,j}\} \setminus \{y_i\}$, where $j \in \{0, \cdots, n-1\}$. We repeat "Step 2" for the set $\{x_{i,j}\}_{j=0}^{n-1}$ to obtain the corresponding representations $\{\bar{Z}_{i,j}\}_{j=0}^{n-1}$. The average of it is denoted by $\bar{Z'}_i$, which serves as a Monte Carlo approximation of the output representation of $\Phi\cdot\pi^*(x)$ for y_i. **Step 4:** Approximating $\Psi\cdot\pi^*(x)+\pi^*(q)$. Combining the previous steps, $\bar{Z}_i - \bar{Z'}_i + \bar{z}$ serves as a Monte Carlo approximation of the output representation of $\Psi \cdot \pi^*(x) + \pi^*(q)$. **Step 5:** Finally, Υ_i is calculated by:

$$\Upsilon_i = \alpha \cdot \cos(z_i, \bar{Z}_i - \bar{Z'}_i + \bar{z}), \tag{11}$$

where α is a scaling hyperparameter, and $\cos(\cdot,\cdot)$ denotes the cosine similarity between two vectors.

Design of KL Divergence. The design of $D_{\text{KL}}(\pi_\theta \parallel \pi'_{\text{ref}})$ is directly inspired by Theorem 1. According to Equation (4), the expected risk of $\pi^*(x)$ is higher than that of $\Psi \cdot \pi^*(x) + \pi^*(q)$. This suggests that the output generated by $\Psi \cdot \pi^*(x) + \pi^*(q)$ may have better quality than the one produced by $\pi^*(x)$. Since π represents the probability distribution over output tokens, while π^* corresponds to the actual generated content, a natural way to improve the performance of π is to encourage its output distribution to align with that of $\Psi \cdot \pi(x) + \pi(q)$. This motivates the use of KL divergence as a regularization term. Specifically, for $D_{\text{KL}}(\pi_\theta \parallel \pi'_{\text{ref}})$, the definition follows a procedure similar to Equation (7):

$$\sum_{i=0}^{n-1}[\frac{\pi'_{\text{ref}}(x_i)}{\pi_\theta(y_{i,j}|x_i, y_{i,<j})} - \log\frac{\pi'_{\text{ref}}(x_i)}{\pi_\theta(y_{i,j}|x_i, y_{i,<j})} - 1], \tag{12}$$

where $\pi'_{\text{ref}}(x_i) = \Psi \cdot \pi(y_{i,j}|x_i, y_{i,<j}) + \pi(y_{i,j}|q, y_{i,<j})$. Based on the analysis in the previous paragraph, we derive the following approximations: (1) $\Phi \cdot \pi(y_{i,j} \mid x_i, y_{i,<j})$ can be approximated by $\sum_{l=0}^{n-1}\pi(y_{i,j} \mid x_i, y_{n,l}, y_{i,<j})$, and (2) the analytical expression of $\Psi \cdot \pi(y_{i,j} \mid x_i, y_{i,<j}) + \pi(y_{i,j} \mid q, y_{i,<j})$ can be approximated by the following equation:

$$\begin{aligned}\pi(y_{i,j} \mid x_i, y_{i,<j}) - \sum_{l=0}^{n-1}\pi(y_{i,j} \mid x_i, y_{n,l}, y_{i,<j})\\+\pi(y_{i,j} \mid q, y_{i,<j}).\end{aligned} \tag{13}$$

Note that, similar to GRPO, the computation in Equation (12) is also carried out in a token-wise manner. Finally, the training process is also similar to GRPO. In the appendix, the overall procedure of the GCPO training is illustrated through the pseudocode.

Experiments

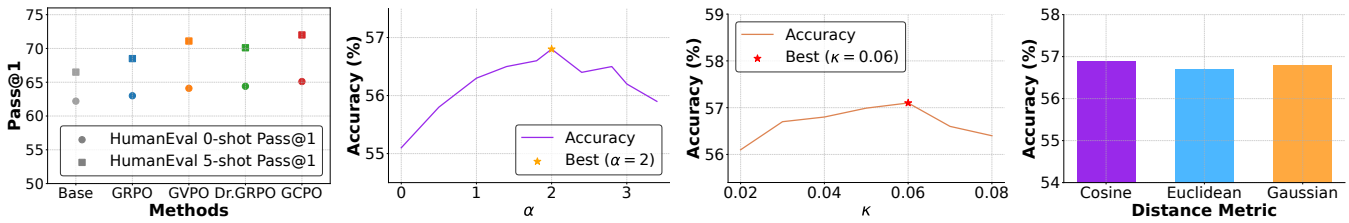
In this section, we conduct comprehensive experiments and ablation studies on multiple reasoning benchmarks to evaluate the effectiveness of our proposed method.

Experimental Settings

We conduct evaluations of our method across several reasoning benchmarks, including AIME24-25, AMC, MATH500

Base model + Method	AIME 2024	AIME 2025	AMC 2023	MATH500	MinervaMATH	**Avg.**
DeepScaleR-1.5B-Preview	42.8	36.7	83.0	85.2	24.6	54.5
+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)	55.0 (+0.5)
+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)	55.4 (+0.9)
+GVPO (Zhang et al. 2025)	46.1 (+3.3)	39.7 (+3.0)	83.6 (+0.6)	85.7 (+0.5)	25.3 (+0.7)	56.1 (+1.6)
+Dr.GRPO (Liu et al. 2025)	45.8 (+3.0)	39.6 (+2.9)	82.1 (-0.9)	85.3 (+0.1)	25.1 (+0.5)	55.6 (+1.1)
+GCPO (Ours)	**46.7 (+3.9)**	**40.3 (+3.6)**	**84.1 (+1.1)**	**86.3 (+1.1)**	**25.9 (+1.4)**	**56.8 (+2.3)**
DeepSeek-R1-Distill-Qwen-1.5B	28.7	26.0	69.9	80.1	19.8	44.9
+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)	46.0 (+1.1)
+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)	46.4 (+1.5)
+GVPO (Zhang et al. 2025)	30.6 (+1.9)	28.2 (+2.2)	71.5 (+1.6)	80.5 (+0.4)	23.1 (+3.3)	46.7 (+1.8)
+Dr.GRPO (Liu et al. 2025)	30.4 (+1.7)	28.4 (+2.4)	71.3 (+1.4)	80.8 (+0.7)	22.9 (+3.1)	46.9 (+2.0)
+GCPO (Ours)	**31.0 (+2.3)**	**29.0 (+3.0)**	**71.8 (+1.9)**	**81.6 (+1.5)**	**23.4 (+3.6)**	**47.4 (+2.5)**
DeepSeek-R1-Distill-Qwen-7B	55.5	50.2	85.1	87.4	42.1	64.1
+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)	65.1 (+1.0)
+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)	65.2 (+1.1)
+GVPO (Zhang et al. 2025)	57.5 (+2.0)	52.1 (+1.9)	86.3 (+1.2)	88.5 (+1.1)	44.2 (+2.1)	65.7 (+1.6)
+Dr.GRPO (Liu et al. 2025)	57.4 (+1.9)	52.3 (+2.1)	86.4 (+1.3)	88.2 (+0.8)	44.0 (+1.9)	65.7 (+1.6)
+GCPO (Ours)	**58.3 (+2.8)**	**53.0 (+2.8)**	**87.3 (+2.2)**	**89.1 (+1.7)**	**45.0 (+2.9)**	**66.5 (+2.4)**

Table 1: Pass@1 performance on various math reasoning benchmarks. We compare base models trained with different fine-tuning approaches. The best results are highlighted in **bold**.



(a) Performance on code tasks. (b) Ablation Study of α (c) Ablation Study of κ (d) Evaluation of similarity metric

Figure 3: Performance analysis and ablation study. (a) shows the performance on code reasoning tasks. (b) and (c) shows the results for parameter sensitivity, i.e., hyperparameter α and κ. (d) shows the evaluation of similarity metric in Υ_i.

(Hendrycks et al. 2021), MinervaMATH (Lewkowycz et al. 2022), and HumanEval (Chen et al. 2021). Our experiments use DeepScaleR-1.5B-Preview and DeepSeek-R1-Distill-Qwen-1.5B, and DeepSeekR1-Distill-Qwen-7B, and Qwen2-7B-Instruct as base models. We compare our method against classic and SOTA reinforcement learning methods, including GRPO (Shao et al. 2024), GVPO (Zhang et al. 2025), ReST-MCTS (Zhang et al. 2024), and Dr.GRPO (Liu et al. 2025). DeepScaleR-1.5B-Preview, having been previously fine-tuned on 40k math QA pairs, is further fine-tuned on 919 AIME problems from 1989 to 2023. DeepSeek-R1-Distill-Qwen-1.5B is fine-tuned on a random subset of 4,000 QA pairs from NuminaMath (Li et al. 2024). Following (Wang et al. 2025a), all training and evaluation stages are constrained to a token budget of 16,384. We adopt a learning rate of $1e{-}6$, weight decay of 0.01, and batch size of 256. All experiments are conducted on A100 GPU clusters.

Performance Analysis

Performance on Mathematical Reasoning Tasks. We evaluate our method against all baseline approaches across a comprehensive set of benchmarks, including AIME 2024, AIME 2025, AMC 2023, MATH500, and MinervaMATH. We compare three widely used base models, including DeepScaleR-1.5B-Preview, DeepSeek-R1-Distill-Qwen-1.5B, and DeepSeek-R1-Distill-Qwen-7B. We report the pass@1 accuracy following (Wang et al. 2025a). Table 1 shows the pass@1 performance. From the results, we can observe that across all settings, GCPO consistently achieves the best average performance, outperforming both the base models and all competitive baselines. Specifically, for DeepScaleR-1.5B-Preview, GCPO delivers an average improvement of 2.3% over the base model. Similar trends are observed for DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B, where GCPO yields average improvements of 2.5% and 2.2%, respectively. Compared to SOTA RL baselines such as GRPO and GVPO, GCPO offers stronger gains on more challenging datasets (e.g., over 1% on AIME and MinervaMATH), highlighting its effectiveness in capturing complex reasoning patterns through causally informed optimization. Notably, the performance margins between GCPO and the strongest baseline

Configurations of GCPO	AIME 2024	AIME 2025	AMC 2023	MATH500	MinervaMATH	Avg.
GCPO (Full)	46.7	40.3	84.1	86.3	25.9	56.8
w/o Adv. Weighting	45.0	39.0	83.0	85.4	24.6	55.4
w/o KL Term	45.3	39.4	83.3	85.3	24.9	55.6

Table 2: Ablation study of the components within GCPO. We report Pass@1 on five benchmarks. Removing either the advantage weighting or KL divergence degrades overall performance, indicating both are essential.

are largest on the hardest benchmarks, indicating that the benefits of causal group structure modeling become more pronounced as task complexity increases. Moreover, the consistent improvements demonstrate that GCPO is robust to model scale and architecture, providing a generalizable fine-tuning strategy for math reasoning tasks.

Performance on Code Reasoning Tasks. Further, for code reasoning tasks, we run different fine-tuning pipelines on Qwen2-7B-Instruct and evaluate them using the standard HumanEval protocol. The results are summarized in Figure 3(a), which reports both 0-shot and 5-shot Pass@1 accuracies for each method. From the results, we can observe that across both evaluation settings, GCPO achieves the strongest performance among all compared methods. In particular, GCPO attains a 0-shot Pass@1 of 65.1% and a 5-shot Pass@1 of 72.0%, outperforming the foundation model by 2.9% and 5.5%, respectively. Relative to other policy optimization methods, such as GRPO and GVPO, GCPO consistently delivers higher accuracy. Notably, the gap between GCPO and previous methods becomes even more pronounced in the multi-shot evaluation, highlighting the advantage of incorporating causal projection and structure-aware regularization in leveraging contextual information and enabling compositional reasoning. These results further demonstrate the effectiveness of the proposed GCPO.

Visualization Analysis Given the substantial computational cost of training LLMs, maintaining stable training dynamics is crucial. To assess this, we use the gradient norm (as a proxy for policy variance) to measure training stability. We record the gradient norm during training for both the baseline methods and the proposed GCPO. The results, presented in the Appendix (Figure 1), demonstrate that GCPO achieves the highest stability, with the gradient norm remaining consistently steady throughout training.

Ablation Study

The effect of different components. To assess the effect of each component in GCPO, we conduct ablation studies. Specifically, we consider two alternative configurations: (i) removing the advantage weighting term (i.e., reusing A_i for all i); and (ii) removing the additional KL divergence term (i.e., setting $\kappa = 0$). Notably, the overall contribution of our reward formulation has already been substantiated in Table 1. Here, we focus on isolating the impact of these individual mechanisms. The ablation results are shown in Table 2. We can observe that both terms are critical for LLM reasoning. These findings underscore the advantages of our design and the effectiveness of GCPO.

Parameter sensitivity. We select the hyperparameters of GCPO based on a systematic evaluation of reasoning performance. We conduct a grid search over the hyperparameter α and the KL regularization coefficient κ to identify the optimal configuration. For α, we explore a range of values: $[0, 0.5, 1, 1.4, 1.8, 2, 2.4, 2.8, 3, 3.4]$. For κ, we first perform a coarse grid search over $[0.02, 0.04, 0.06, 0.08]$ with a step size of 0.02, and subsequently conduct a finer search within the promising interval using a step size of 0.01. For each configuration of (α, β), we record the Pass@1 performance. Figure 3(b)-(c) show that model accuracy initially increases with larger values of α and κ, but plateaus or slightly degrades when these values become too large; the best results are consistently achieved with $\alpha = 2$ and $\kappa = 0.06$. These values are thus adopted as our hyperparameter settings.

Evaluation of metric for Υ_i According to Eq. 11, we compute Υ_i by calculating the cosine similarity $cos(\cdot)$. To evaluate the impact of this metric on performance, we conduct an ablation study comparing different similarity measures, including cosine similarity, Euclidean distance, and Gaussian distance. The results are shown in Figure 3(d). With the introduction of the hyperparameter α, the performance differences among various similarity measures are negligible. We ultimately select cosine similarity as the default metric, primarily because it allows for more flexible and convenient tuning of α (See the Appendix for details).

Conclusion

In this paper, we present GCPO, a novel post-training method that integrates causal structure into policy optimization for large language models. Building on the limitations of GRPO, GCPO addresses the overlooked interdependencies among groupwise candidate responses by modeling them through an SCM. Our analysis reveals that conditioning on a final integrated response induces a collider structure, which in turn exposes latent dependencies among originally independent candidates. Guided by this insight, GCPO introduces two key components: (1) a causally-adjusted reward mechanism that projects individual responses onto a structurally coherent subspace, and (2) a KL-divergence regularization term that aligns the policy with a causally-informed reference distribution. Extensive experiments across multiple reasoning benchmarks demonstrate that GCPO substantially outperforms existing baselines, confirming the benefits of incorporating causal reasoning into groupwise optimization. Our findings underscore the importance of structural awareness in reinforcement learning for LLM post-training and suggest promising directions for future work on causality-aware RLHF.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China under Grants 62506355 and 42506186, and the numerical calculations in this study were partially performed on the ORISE Supercomputer (DFZX202416).

References

Bai, H.; Zhou, Y.; Cemri, M.; Pan, J.; Suhr, A.; Levine, S.; and Kumar, A. 2024. DigiRL: Training In-The-Wild Device-Control Agents with Autonomous Reinforcement Learning. arXiv:2406.11896.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *arXiv preprint arXiv:2204.05862*.

Ballon, M.; Algaba, A.; and Ginis, V. 2025. The Relationship Between Reasoning and Performance in Large Language Models–o3 (mini) Thinks Harder, Not Longer. *arXiv preprint arXiv:2502.15631*.

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb): 1137–1155.

Bottou, L.; Curtis, F. E.; and Nocedal, J. 2018. Optimization Methods for Large-Scale Machine Learning. arXiv:1606.04838.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. D. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Cheng, Z.; Chen, D.; Fu, M.; and Zhou, T. 2025. Optimizing Length Compression in Large Reasoning Models. *arXiv preprint arXiv:2506.14755*.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, E.; Wang, X.; Hu, X.; Roberts, A.; Mehta, H.; Wei, J.; Chandu, K. R.; Gritsenko, A.; Piantanida, P.; Chowdhery, A.; Clark, J. H.; Schick, T.; Dwivedi-Yu, J.; Yu, J.; Shi, K.; Li, X.; Ippolito, D.; Zhou, D.; Ainslie, J.; Firat, O.; Lu, Y.; Dean, J.; Le, Q. V.; and Chi, E. H. 2022. Scaling Instruction-Finetuned Language Models. *arXiv preprint arXiv:2210.11416*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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)*, 4171–4186.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep learning*, volume 1.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Jaech, A.; Kalai, A.; Lerer, A.; Richardson, A.; El-Kishky, A.; Low, A.; Helyar, A.; Madry, A.; Beutel, A.; Carney, A.; et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Lai, H.; Liu, X.; Gao, J.; Cheng, J.; Qi, Z.; Xu, Y.; Yao, S.; Zhang, D.; Du, J.; Hou, Z.; et al. 2025. A Survey of Post-Training Scaling in Large Language Models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2771–2791.

Lewkowycz, A.; Andreassen, A.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35: 3843–3857.

Li, J.; Beeching, E.; Tunstall, L.; Lipkin, B.; Soletskyi, R.; Huang, S.; Rasul, K.; Yu, L.; Jiang, A. Q.; Shen, Z.; et al. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13: 9.

Liu, Z.; Chen, C.; Li, W.; Qi, P.; Pang, T.; Du, C.; Lee, W. S.; and Lin, M. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Minaee, S.; Mikolov, T.; Nikzad, N.; Chenaghlu, M.; Socher, R.; Amatriain, X.; and Gao, J. 2024a. Large Language Models: A Survey. *arXiv preprint arXiv:2402.06196*.

Minaee, S.; Mikolov, T.; Nikzad, N.; Chenaghlu, M.; Socher, R.; Amatriain, X.; and Gao, J. 2024b. Large language models: a survey (2024). *URL https://arxiv. org/abs/2402.06196*, 7(8): 9.

Muennighoff, N.; Yang, Z.; Shi, W.; Li, X. L.; Fei-Fei, L.; Hajishirzi, H.; Zettlemoyer, L.; Liang, P.; Candès, E.; and Hashimoto, T. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

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

Pearl, J.; Glymour, M.; and Jewell, N. P. 2016. *Causal inference in statistics: A primer*. John Wiley & Sons.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Sadik, A. R.; and Govind, S. 2025. Benchmarking LLM for Code Smells Detection: OpenAI GPT-4.0 vs DeepSeek-V3. arXiv:2504.16027.

Sanh, V.; Webson, A.; Raffel, C.; Bach, S. H.; Sutawika, L.; Alyafeai, Z.; Chaffin, A.; Srinivasan, A.; Yong, Z. X.; Kim, T.; Crowell, E. S.; Kudugunta, S.; Sharma, A.; Ong, R.; Sharma, S.; Lo, A.; Bari, M. S.; Xu, C.; Thakker, U.;

Dey, M.; Desai, S.; Sangwan, R.; Geng, X.; Arora, D.; Ram, D.; Wang, H.; Chandu, K.; Kashyap, A.; Tan, S.; Gotmare, A. D.; Swabha, S.; Phang, J.; Chan, H. P.; Urbanek, J. H.; Gururangan, S.; d. S. Clemente, M. V.; McMahan, B.; Albanie, S.; Welbl, J.; Liu, Q.; Malmi, E.; Jean, S.; Kuo, J. T.; Jiang, M. T.-J.; Xu, Y.; Conneau, A.; McCoy, R. T.; Taylor, S.; Smith, N. A.; Zettlemoyer, L.; Ruder, S.; Yogatama, D.; Cho, K.; and Rush, A. M. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *International Conference on Learning Representations (ICLR)*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Sun, C.; He, P.; Ji, Q.; Zang, Z.; Li, J.; Wang, R.; and Wang, W. 2024a. M2i2: Learning efficient multi-agent communication via masked state modeling and intention inference. *arXiv preprint arXiv:2501.00312*.

Sun, C.; He, P.; Wang, R.; and Zheng, C. 2025. Revisiting Communication Efficiency in Multi-Agent Reinforcement Learning from the Dimensional Analysis Perspective. In Das, S.; Nowé, A.; and Vorobeychik, Y., eds., *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2025, Detroit, MI, USA, May 19-23, 2025*, 1977–1986. International Foundation for Autonomous Agents and Multiagent Systems / ACM.

Sun, C.; Wu, B.; Wang, R.; Hu, X.; Yang, X.; and Cong, C. 2021. Intrinsic Motivated Multi-Agent Communication. AAMAS '21, 1668–1670. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450383073.

Sun, C.; Zang, Z.; Li, J.; Li, J.; Xu, X.; Wang, R.; and Zheng, C. 2024b. T2mac: Targeted and trusted multi-agent communication through selective engagement and evidence-driven integration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 15154–15163.

Tie, G.; Zhao, Z.; Song, D.; Wei, F.; Zhou, R.; Dai, Y.; Yin, W.; Yang, Z.; Yan, J.; Su, Y.; et al. 2025. A survey on post-training of large language models. *arXiv e-prints*, arXiv–2503.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

Wang, J.; Qiang, W.; Song, Z.; Zheng, C.; and Xiong, H. 2025a. Learning to Think: Information-Theoretic Reinforcement Fine-Tuning for LLMs. arXiv:2505.10425.

Wang, N.; Yao, B.; Zhou, J.; Hu, Y.; Wang, X.; Guan, N.; and Jiang, Z. 2025b. Insights from Verification: Training a Verilog Generation LLM with Reinforcement Learning with Testbench Feedback. arXiv:2504.15804.

Zang, Z.; Sun, C.; Liu, L.; Sun, F.; and Zheng, C. 2025. Loss of Plasticity: A New Perspective on Solving Multi-Agent Exploration for Sparse Reward Tasks. In Das, S.; Nowé, A.; and Vorobeychik, Y., eds., *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2025, Detroit, MI, USA, May 19-23, 2025*, 2299–2308. International Foundation for Autonomous Agents and Multiagent Systems / ACM.

Zhang, D.; Zhoubian, S.; Hu, Z.; Yue, Y.; Dong, Y.; and Tang, J. 2024. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37: 64735–64772.

Zhang, K.; Hong, Y.; Bao, J.; Jiang, H.; Song, Y.; Hong, D.; and Xiong, H. 2025. GVPO: Group variance policy optimization for large language model post-training. *arXiv preprint arXiv:2504.19599*.

Zhang, K. W.; and Bowman, S. R. 2018. Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis. *arXiv preprint arXiv:1809.10040*.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2025. A survey of large language models.