Tree-OPO: Off-policy Monte Carlo Tree-Guided Advantage Optimization for Multistep Reasoning

Bingning Huang[1,*†] **Tu Nguyen**[2,*] **Matthieu Zimmer**[3]

[1]Technical University of Munich
[2]Huawei R&D Munich
[3]Huawei Noah's Ark Lab
bingning.huang@tum.de ✉{tu.nguyen, matthieu.zimmer}@huawei.com

Abstract

Recent advances in reasoning with large language models (LLMs) have shown the effectiveness of Monte Carlo Tree Search (MCTS) for generating high-quality intermediate trajectories, particularly in math and symbolic domains. Inspired by this, we explore how MCTS-derived trajectories—traditionally used for training value or reward models—can be repurposed to improve policy optimization in preference-based reinforcement learning (RL). Specifically, we focus on Group Relative Policy Optimization (GRPO), a recent algorithm that enables preference-consistent policy learning without value networks. We reframe GRPO into a staged training paradigm, leveraging a teacher's MCTS rollouts to construct a tree-structured curriculum of prefixes. This introduces the novel challenge of computing advantages for training samples that originate from different prefixes, each with a distinct expected return. To address this, we propose Staged Advantage Estimation (SAE), a framework for computing low-variance, prefix-aware advantages by projecting rewards onto a constraint set that respects the tree's hierarchy. Our empirical results show that SAE improves final accuracy over standard GRPO on mathematical reasoning tasks. This finding is supported by our theoretical analysis—which proves SAE reduces gradient variance for improved sample efficiency—and is demonstrated using both efficient heuristics and a formal quadratic program.

1 Introduction

Many reasoning-intensive tasks—from mathematical problem-solving to symbolic analysis in security domains like deobfuscation and reverse engineering—require multi-hop, compositional inference [Chr+23]. While Monte Carlo Tree Search (MCTS) has been widely used in structured decision-making—most notably in game-playing [Che16]—recent work has shown its utility in language-based reasoning as well [Qi+24; Luo+25; Wan+24; Zha+24]. In these settings, MCTS systematically explores the reasoning space and balances exploration with exploitation, enabling it not only to improve test-time inference but also to identify reliable reasoning chains. These high-quality trajectories are commonly used to supervise downstream training, typically by training reward models [Qi+24; Luo+25; Wan+24] or enabling self-training from model-generated traces [Zha+25; Zha+24]. In some cases, a strong teacher model guides MCTS rollouts, effectively distilling reasoning behavior into downstream models via these enriched trajectories [Jia+24].

We draw inspiration from this line of work, but take a distinct path: rather than using MCTS *rollouts* solely for supervised reward modeling or test-time inference, we repurpose *partial trajectories*,

[*]Equal contribution.
[†]Interned at the AI4Sec team, Huawei Munich Research Center.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

generated by a strong teacher policy, to construct enriched training groups for preference-based policy optimization. In particular, we examine Group Relative Policy Optimization (GRPO) [Dee+25], a method that can be broadly categorized as *preference-based* RL in the sense that it learns from the relative quality of samples within a group rather than relying on an explicit value function.

Our main observation is that the staged nature of MCTS rollouts naturally induces a **tree-structured space of prompt-completion pairs** [3], where each path from root to leaf corresponds to a sequence of reasoning steps. This setting introduces a previously unstudied challenge in GRPO: how to compute meaningful advantages when group samples correspond to different prefixes in a shared trajectory. To address this, we introduce *Staged Advantage Estimation (SAE)*, a method for computing advantages across this structured curriculum by accounting for prefix-conditioned values.

We evaluate the staged GRPO setting on mathematical reasoning tasks, currently training primarily on GSM8K [Cob+21], and observe measurable improvements over standard GRPO. These gains arise from our SAE, which explicitly exploits the prefix-conditioned structure of MCTS rollouts to compute more informative advantage estimates. Theoretical analysis shows that stage-wise baselines reduce gradient variance and align with conditional success probabilities, providing principled guidance for advantage computation. While SAE performs robustly, the magnitude of improvements is naturally limited by objective factors such as model capacity and dataset difficulty, suggesting that richer tree-structured supervision could yield larger benefits on more challenging mathematical reasoning tasks. Our work highlights new directions in integrating structured trajectory supervision with RL methods and opens up theoretical questions about policy learning in tree-structured domains.

2 Methodology

We extend GRPO, a method that relies on on-policy rollouts where the target policy must rediscover entire reasoning paths from a single prompt. This standard approach can be sample-inefficient for complex tasks due to sparse rewards and challenging exploration. Our key insight is to replace this on-policy sampling with a more structured, off-policy curriculum. We use a strong teacher policy to perform MCTS offline, generating a rich dataset of valuable partial trajectories (prefixes). The student then learns by starting its rollouts from these informative, teacher-vetted reasoning states.

This reframes the learning problem from solving a single hard task to mastering a curriculum of diverse, more tractable sub-problems. By decomposing each MCTS rollout into a tree of prefixes, we achieve significant variance reduction and provide a denser, more targeted learning signal, yielding our Tree-structured Off-policy Optimization (Tree-OPO) framework.

Learning from a Curriculum of MCTS-Derived Prefixes. Our Tree-OPO framework operationalizes this curriculum by sampling prefixes p from the teacher's MCTS-derived tree (D) at each update, from which the student policy π_θ generates a continuation $\hat{c}$ to receive a sparse, terminal reward r. This prefix tree naturally embodies an implicit **reverse curriculum**: prefixes represent subproblems of varying difficulty, providing a denser and more diverse learning signal than sampling full trajectories from scratch. This staged setting, however, requires a more sophisticated advantage signal for the policy gradient update. While the ideal, variance-minimizing baseline is the true prefix-value $V(p) = \mathbb{E}[r \mid p]$, estimating this directly for every prefix via rollouts is computationally expensive and high-variance. We therefore approximate this ideal baseline by leveraging the tree's structural regularities.

Staged Advantage Estimation (SAE). In standard GRPO, advantages are derived by mean-centering rewards, $A_k = r_k - \bar{r}$; a subsequent, optional normalization by standard deviation is irrelevant to the core issue here. This mean-centering is effective when all trajectories share a single prompt, as their true expected returns $V^\pi(p)$ are identical. Our Tree-OPO setting operates outside this assumption: a training batch contains samples from diverse prefixes $\{p_k\}$ with disparate true values. Applying a single mean baseline to rewards with different underlying expectations creates a systematically biased advantage signal, which inflates gradient variance and confounds credit assignment between easy (deep) and hard (shallow) prefixes. (Fig. 1, top).

To this end, we propose the Staged Advantage Estimation (SAE), which formulates the advantage calculation as a constrained optimization problem. Given a group of N rewards $\boldsymbol{r}$, SAE finds an

[3] We use "staged" and "tree-structured" interchangeably for this hierarchical setting.

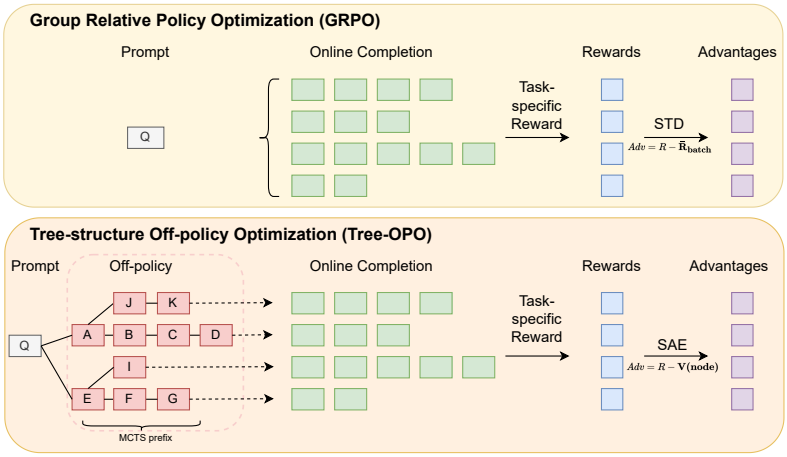


Figure 1: **SAE vs. GRPO.** In contrast to GRPO's flat standardization (top), SAE in the Tree-OPO framework (bottom) uses a constrained optimization to produce structured, prefix-aware advantages that respect the MCTS-derived hierarchy.

advantage vector $\boldsymbol{a}$ that is closest to $\boldsymbol{r}$ while satisfying MCTS-derived structural constraints:

$$\begin{aligned} \min_{\mathbf{a}\in\mathbb{R}^N} \quad & \|\mathbf{a}-\mathbf{r}\|^2, \\ \text{s.t.} \quad & \mathbf{1}^\top\mathbf{a}=0, \;\; \|\mathbf{a}\|^2 \le N, \;\; a_i+\delta_{ij}\le a_j \quad \forall\,(i,j)\in\mathcal{C}_{\text{order}}. \end{aligned} \tag{1}$$

This presents the **soft**-constraint or **convex** formulation of SAE, which serves as the basis for our theoretical analysis. Its objective QP objective $\|\boldsymbol{a}-\boldsymbol{r}\|^2$ anchors the solution to the empirical rewards, while the crucial $\mathcal{C}_{\text{order}}$ constraint provides a structural inductive bias by enforcing **tree consistency**. These constraints go beyond simple value-shaping: they rank parent-child pairs based on success and systematically re-weight siblings to bias exploration toward less-proven but promising branches (see App. B.1). When formulated as a convex problem by relaxing the normalization to an inequality ($\|\boldsymbol{a}\|^2 \le N$), our theoretical analysis (Thm. B.4) shows this approach guarantees benefits like variance reduction. Our practical evaluation compares two main strategies. The primary approach uses computationally efficient heuristic baselines, which compute advantages as $a'_i = r_i - \alpha\, V(p_i)$ based on either the empirical **Expectation** (subtree success rate), an **Optimistic** value for exploration, or a **Pessimistic** value for conservative updates (see further B.2). To benchmark these heuristics against structured estimation, we include both the **SAE (Hard)** variant, which enforces the non-convex normalization ($\|\boldsymbol{a}\|^2 = N$), and the **SAE (Soft)** variant with the relaxed convex formulation—found to be significantly more stable and consistently superior in practice.

Heuristic baseline estimators. While the SAE (Soft) formulation provides the most principled and stable estimate, it is computationally heavier than simple analytic baselines. To complement it, we introduce three tree-consistent heuristic estimators for $V(p)$ that approximate the structure-aware value of the SAE while remaining lightweight and easily interpretable:

1. **Empirical (Expectation):**

$$V_E(p) = \frac{\#\{\text{successful rollouts from } p\}}{\#\{\text{total rollouts from } p\}},$$

the subtree success rate. By Lemma B.2, the ideal baseline $V^*(p) = \mathbb{E}[r \mid p]$ uniquely minimizes variance and maximizes reward alignment. $V_E(p)$ is its natural Monte–Carlo approximation. Combined with mean-centering, $a'_i = r_i - \alpha V_E(p_i)$, this yields tree-consistent advantages with near-optimal variance reduction.

2. **Optimistic:**

$$V_O(p) = \mathbf{1}\{\exists\,\text{successful rollout in } p\text{'s subtree}\},$$

which amplifies sparse positive signals and encourages exploration.

Method	Training Data	Advantage	$V(x)$	Acc. (%)	Constraint Sat. (%)
GRPO [Sha+24]	GSM8K	Flat	–	76.27	–
Tree-OPO	GSM8K-MCTS	Flat	–	75.66	79.26 ± 31.94
		Trace	–	73.91	–
		Tree (Heuristic)	Expectation	**77.63**	97.98 ± 5.59
			Optimistic	70.58	–
			Pessimistic	67.40	–
		SAE (*Hard*)	–	75.21	100 ± 0
		SAE (*Soft*)	–	77.41	100 ± 0

Table 1: Comparison of GRPO and Tree-OPO variants on GSM8K. *Constraint Sat.* measures the percentage of ordering constraints satisfied by raw rewards before applying SAE. A dash (–) denotes non-applicability or omission for auxiliary variants.

3. **Pessimistic:**

$$V_P(p) = \mathbf{1}\{\nexists \text{ failed rollout in } p\text{'s subtree}\},$$

promoting conservative updates by penalizing uncertainty in downstream paths.

These heuristics capture different assumptions about future trajectories while remaining tree-consistent and computationally light. Together with mean-centering, they approximate the variance- and reward-alignment properties of the ideal solution to (2), offering a practical substitute for exact constrained optimization. Overall, the Empirical (Expectation) baseline tends to deliver the best balance between variance reduction and reward alignment, while the Optimistic and Pessimistic variants are more suitable in sparse-reward or high-stakes settings where exploration or conservativeness is preferred.

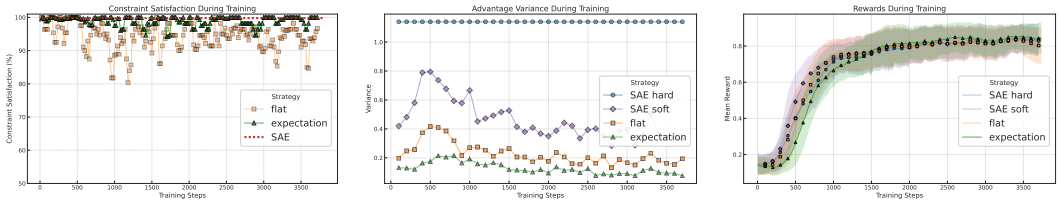
3 Experiment

Setup. We train models on **GSM8K** [Cob+21], a standard benchmark for multi-step mathematical reasoning, and evaluate them on GSM8K, **GSM-Symbolic** [Mir+24] and **MATH** [Hen+21] to provide robust performance estimates and reduce reward exploitation. For staged learning, we employ `GSM8K-MCTS`, an offline dataset generated via the MCTS-driven multi-turn chain-of-thought (CoT) procedure detailed in [Qi+24]. This dataset comprises tree-structured reasoning trajectories derived from 16 MCTS rollouts per problem, each with a maximum depth of 5 steps and up to 5 child nodes explored per step. These trajectories are *augmented* with all their prefixes (e.g., ABCDE yields A, AB, ABC, ABCD), resulting in approx. 160k unique staged prefixes. Experiments employ `Qwen2.5-1.5B` as the student policy and `Qwen2.5-7B` as the teacher when applicable. The final answer accuracies (pass@1) on the test sets are reported.

Advantage estimation structures. We compare three structures for advantage estimation: *flat* (vanilla per-step, no hierarchy), *trace* (advantages along single rollouts without interleaving, used as ablation), and *tree* (models full MCTS prefix hierarchy).

Results. Tree-structured advantage estimation improves over both flat and trace-based baselines, achieving 77.63% accuracy under the expectation heuristic. This setup best respects the MCTS-derived prefix hierarchy and yields the most informative gradient signal. Among the baselines, the expectation-based $V(x)$ outperforms both **optimistic** and **pessimistic** variants, suggesting that the empirical values best satisfy the ranking constraints and stabilize updates. While the gains over vanilla GRPO are modest, the comparison between Tree-OPO variants is more revealing. The expectation-based baseline achieves the highest accuracy, a result strongly correlated with its low advantage variance and nearly perfect tree-consistency constraint satisfaction (97.98%, as shown in Figures 3b and 3a).

In contrast, the **SAE (Hard)** variant serves as an ablative baseline testing the effect of enforcing perfect constraint satisfaction. Its non-convex normalization ($\|\boldsymbol{a}\|^2 = N$) fixes the advantage variance near 1.0, distorting the learning signal by decoupling it from the true reward scale. This rigidity leads to instability and degraded performance (**75.21%**).



(a) Constraint satisfaction (b) Advantage variance (c) Group average reward

Figure 3: **Training metrics over time.** (a) Satisfaction rate of prefix-ordering constraints imposed by structured advantage estimation. (b) Variance of advantage estimates; lower variance improves update stability. (c) Task-specific average rewards during training.

.

Relaxing the constraint in the **SAE (Soft)** formulation ($\|\boldsymbol{a}\|^2 \leq N$) yields a smoother balance between structural consistency and adaptive scaling, improving stability and performance to **77.41%**. This places it on par with the best-performing **Expectation** heuristic (**77.63%**), which benefits from its unbiased variance-minimizing nature (Lemma B.2). The convex SAE thus matches the leading heuristic while retaining theoretical guarantees of *gradient* variance reduction (Thm. B.4) and the interpretability of a structured projection. Empirically, the observed *advantage* variance of SAE (Soft) can exceed that of Tree-OPO (Flat) at certain steps. This does not contradict the theory, which concerns the variance of the policy-gradient estimator under projections of the same reward vector. In practice, the soft formulation introduces relaxed penalties and numerical tolerances that can yield slightly higher apparent advantage variance when the norm constraint is not tightly enforced. Such deviations stem from implementation and tuning effects rather than a failure of the variance-reduction guarantee. Nevertheless, SAE consistently preserves the non-increasing variance behavior relative to its own projected inputs, validating the theoretical result in practice. The modest gains on GSM8K likely reflect the dataset's simple tree structures, which limit the benefit of more sophisticated structural estimators.

Cross-Dataset Performance. Figure 2 presents a comparison of the approaches' performance across different math-reasoning benchmarks. The results demonstrate the robustness of our methods while also showing that GRPO is a strong baseline, particularly when its global advantages naturally satisfy the tree-consistency constraints.

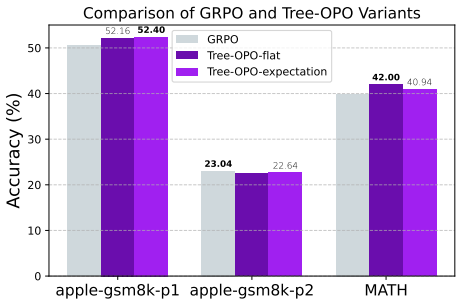


Figure 2: Performance comparison across different datasets.

4 Conclusion

We introduced **Tree-OPO**, a framework that repurposes MCTS trajectories into a structured curriculum to improve preference-based policy optimization. Our method, **Staged Advantage Estimation (SAE)**, leverages the tree's hierarchy to compute low-variance, prefix-aware advantages. On mathematical reasoning tasks, this approach improves sample efficiency and outperforms unstructured baselines. Importantly, Tree-OPO is not merely another tree-based optimizer: by using offline teacher MCTS prefixes in an off-policy curriculum, it addresses a distinct regime where heterogeneous prefix expectations arise, and SAE provides a principled solution.

References

[Aga+24] R. Agarwal, N. Vieillard, Y. Zhou, P. Stanczyk, S. R. Garea, M. Geist, and O. Bachem. "On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes." In: *The Twelfth International Conference on Learning Representations*. 2024.

[BC17] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. 2nd. Springer, 2017.

[Che16] J. X. Chen. "The Evolution of Computing: AlphaGo." In: *Computing in Science & Engineering* 18.4 (2016), pp. 4–7. DOI: 10.1109/MCSE.2016.74.

[Chr+23] F. Christianos, G. Papoudakis, M. Zimmer, T. Coste, Z. Wu, J. Chen, K. Khandelwal, J. Doran, X. Feng, J. Liu, et al. "Pangu-agent: A fine-tunable generalist agent with structured reasoning." In: *arXiv preprint arXiv:2312.14878* (2023).

[Cob+21] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. *Training Verifiers to Solve Math Word Problems*. 2021. arXiv: 2110.14168 [cs.LG].

[Dee+25] DeepSeek-AI, D. Guo, D. Yang, et al. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. arXiv: 2501.12948 [cs.CL].

[GBB04] E. Greensmith, P. L. Bartlett, and J. Baxter. "Variance reduction techniques for gradient estimates in reinforcement learning." In: *Journal of Machine Learning Research*. 2004.

[Gu+24] Y. Gu, L. Dong, F. Wei, and M. Huang. *MiniLLM: Knowledge Distillation of Large Language Models*. 2024. arXiv: 2306.08543 [cs.CL].

[Hen+21] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. "Measuring Mathematical Problem Solving With the MATH Dataset." In: *NeurIPS* (2021).

[Hsi+23] C.-Y. Hsieh, C.-L. Li, C.-K. Yeh, H. Nakhost, Y. Fujii, A. Ratner, R. Krishna, C.-Y. Lee, and T. Pfister. *Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes*. 2023. arXiv: 2305.02301 [cs.CL].

[Jia+24] J. Jiang, Z. Chen, Y. Min, J. Chen, X. Cheng, J. Wang, Y. Tang, H. Sun, J. Deng, W. X. Zhao, et al. "Enhancing llm reasoning with reward-guided tree search." In: *arXiv preprint arXiv:2411.11694* (2024).

[Ko+24] J. Ko, S. Kim, T. Chen, and S.-Y. Yun. *DistiLLM: Towards Streamlined Distillation for Large Language Models*. 2024. arXiv: 2402.03898 [cs.CL].

[Ko+25] J. Ko, T. Chen, S. Kim, T. Ding, L. Liang, I. Zharkov, and S.-Y. Yun. *DistiLLM-2: A Contrastive Approach Boosts the Distillation of LLMs*. 2025. arXiv: 2503.07067 [cs.CL].

[Li+25] Y. Li, Q. Gu, Z. Wen, Z. Li, T. Xing, S. Guo, T. Zheng, X. Zhou, X. Qu, W. Zhou, Z. Zhang, W. Shen, Q. Liu, C. Lin, J. Yang, G. Zhang, and W. Huang. *TreePO: Bridging the Gap of Policy Optimization and Efficacy and Inference Efficiency with Heuristic Tree-based Modeling*. 2025. arXiv: 2508.17445 [cs.LG].

[Liu+25] Z. Liu, C. Chen, W. Li, P. Qi, T. Pang, C. Du, W. S. Lee, and M. Lin. *Understanding R1-Zero-Like Training: A Critical Perspective*. 2025. arXiv: 2503.20783 [cs.LG].

[Luo+25] L. Luo, Y. Liu, R. Liu, S. Phatale, M. Guo, H. Lara, Y. Li, L. Shu, L. Meng, J. Sun, and A. Rastogi. *Improve Mathematical Reasoning in Language Models with Automated Process Supervision*. 2025.

[Mir+24] I. Mirzadeh, K. Alizadeh, H. Shahrokhi, O. Tuzel, S. Bengio, and M. Farajtabar. *GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models*. 2024.

[Qi+24] Z. Qi, M. Ma, J. Xu, L. L. Zhang, F. Yang, and M. Yang. *Mutual Reasoning Makes Smaller LLMs Stronger Problem-Solvers*. 2024. arXiv: 2408.06195 [cs.CL].

[Sha+24] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, and D. Guo. *DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models*. 2024. arXiv: 2402.03300 [cs.CL].

[SW25] K. Saadi and D. Wang. "TASKD-LLM: Task-Aware Selective Knowledge Distillation for LLMs." In: *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*. 2025.

[Wan+24] P. Wang, L. Li, Z. Shao, R. X. Xu, D. Dai, Y. Li, D. Chen, Y. Wu, and Z. Sui. *Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations*. 2024. arXiv: 2312.08935 [cs.AI].

[Wil92] R. J. Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning." In: *Machine Learning*. 1992.

[Yan+25] Z. Yang, Z. Guo, Y. Huang, X. Liang, Y. Wang, and J. Tang. *TreeRPO: Tree Relative Policy Optimization*. 2025. arXiv: 2506.05183 [cs.LG].

[Zha+24] D. Zhang, S. Zhoubian, Z. Hu, Y. Yue, Y. Dong, and J. Tang. *ReST-MCTS*: LLM Self-Training via Process Reward Guided Tree Search*. 2024. arXiv: 2406.03816 [cs.CL].

[Zha+25] Y. Zhai, H. Liu, Z. Zhang, T. Lin, K. Xu, C. Yang, D. Feng, B. Ding, and H. Wang. "Empowering Large Language Model Agent through Step-Level Self-Critique and Self-Training." In: *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '25. Padua, Italy: Association for Computing Machinery, 2025, pp. 1444–1454. ISBN: 9798400715921. DOI: 10.1145/3726302.3729965.

[Zhe+25] C. Zheng, S. Liu, M. Li, X.-H. Chen, B. Yu, C. Gao, K. Dang, Y. Liu, R. Men, A. Yang, J. Zhou, and J. Lin. *Group Sequence Policy Optimization*. 2025. arXiv: 2507.18071 [cs.LG].

A Related Works

LLM reasoning and distillation has been widely explored in literature, though often in isolation. Approaches like MiniLLM [Gu+24], DistiLLM [Ko+24; Ko+25] and GKD [Aga+24] focus on general distillation, typically leveraging Kullback-Leibler (KL) divergence on logits to transfer universal capabilities or factual knowledge. However, these methods often lack a direct focus on distilling intricate reasoning processes, instead concentrating on broader knowledge transfer. More specialized distillation efforts, such as [SW25; Hsi+23], aim for task-specific knowledge transfer, with the latter even targeting intermediate thought processes (rationales). While these works delve into transferring knowledge about *how* to reason (e.g., specific steps), they primarily focus on reproducing teacher-generated content rather than transferring the underlying *skill* of exploring and evaluating reasoning paths.

Separately, reasoning-based RL paradigms, such as GRPO [Sha+24; Zhe+25; Liu+25] and, in particular, its tree-based variants TreeRPO [Yan+25] and TreePO [Li+25], address complex problems by optimizing policies over tree-structured action spaces. However, these methods typically operate in an online, *on-policy* regime, where the learner generates full trajectories from the root and advantages are computed under the assumption that all samples in a group share the same prefix distribution. This design avoids the need to correct for heterogeneous baselines, since every trajectory starts from a common root state. In contrast, our approach operates in a unique **hybrid setting**: we leverage partial trees generated offline by a teacher's policy (e.g., via MCTS) and then require the student policy to complete these sub-trajectories online. This curriculum-driven process induces a *novel tree-structured advantage space*, where samples originate from diverse prefixes with different expected returns. Correctly handling these heterogeneous baselines motivates our SAE, which explicitly incorporates tree consistency into advantage computation. To our knowledge, our work represents the first comprehensive exploration into improving complex reasoning skills in LLMs by actively studying the structure of the advantage space beyond flat or simple group-wise considerations. This offers a distinct and complementary approach to cultivating sophisticated reasoning abilities.

B Extended Methodology

B.1 Staged Advantage Estimation as Constrained Quadratic Program

In standard GRPO, a single baseline (the group mean) suffices because all completions share the same prompt context. However, in our Tree-OPO setting each sample in a group may originate from a different prefix p_k of the MCTS-generated reasoning tree, and these prefixes have *disparate* expected returns. Without correcting for this, shallow prefixes (low $\mathbb{E}[r \mid p]$) and deep prefixes (high $\mathbb{E}[r \mid p]$) are unfairly compared, leading to:

- **Misaligned baselines:** Using a single mean ignores each prefix's own expected return $\mathbb{E}[r \mid q_i]$, leading to biased advantages.
- **Increased variance:** Mixing prefixes of different depths without context-specific centering amplifies gradient noise and hinders convergence.
- **Erroneous credit assignment:** Without tree-aware baselines, a strong outcome on an easy (deep) prefix can overshadow genuine improvements on a harder (shallow) prefix, and v.v.

To address these issues, we formulate advantage estimation as the solution of a constrained quadratic program over the group of staged samples. For a group of N staged pairs $\{(p_k, r_k)\}$ with binary rewards $r_k \in \{0, 1\}$, let $\boldsymbol{a} = (a_1, \ldots, a_N)^T$ be the vector of advantages and $\boldsymbol{r} = (r_1, \ldots, r_N)^T$ be the vector of rewards. The optimization problem is:

$$\begin{aligned} \min_{\mathbf{a}\in\mathbb{R}^N} \quad & \|\mathbf{a} - \mathbf{r}\|^2, \\ \text{s.t.} \quad & \mathbf{1}^\top \mathbf{a} = 0, \\ & \|\mathbf{a}\|^2 \le N, \\ & a_i + \delta_{ij} \le a_j \quad \forall\, (i,j) \in \mathcal{C}_{\text{order}}, \end{aligned} \tag{2}$$

Here, $\delta_{ij} \ge 0$ acts as a margin parameter in the ordering constraints, requiring a_j to exceed a_i by at least δ_{ij}. In our theoretical analysis, we set $\delta_{ij} = 0$ to retain convexity and feasibility, while in practice small positive margins can be introduced to improve robustness to noise.

The first two constraints impose GRPO-style normalization (zero-mean and bounded root-mean-square for a vector of N elements). The final constraint encodes **tree consistency** by enforcing specific orderings on advantage values based on their structural relationships within the MCTS tree. The set $\mathcal{C}_{\text{order}}$ is defined as the union of two distinct sets of pairwise ordering constraints: $\mathcal{C}_{\text{order}} = \mathcal{C}_{\text{pair}} \cup \mathcal{C}_{\text{triplet}}$.

To formally define these constraint sets, let $\mathcal{D}$ denote the entire set of MCTS-generated traces for the current group. We introduce the following predicate functions:

- *IsPrefix*(i, j): True if $p_i \subset p_j$.
- *IsSibling*(i, j): True if p_i and p_j share the same immediate parent prefix.
- *HasSuccessfulContinuation*(p_k): True if an online rollout $\hat{c}$ from π_θ forms a complete, successful trace $\tau_m = p_k \cdot \hat{c}$ (i.e., its binary reward $r_m = 1$). For brevity, we denote this as $S(p_k)$ or $S(k)$ for trace p_k.

Pair-wise (Parent-Child) Consistency Constraints ($\mathcal{C}_{\text{pair}}$) These constraints ensure advantages respect basic prefix containment when a failing prefix is extended into a successful trace:

$$\mathcal{C}_{\text{pair}} = \{(i, j) \mid \textit{IsPrefix}(i, j),\ r_i = 0,\ r_j = 1\}.$$

Triplet Consistency Constraints ($\mathcal{C}_{\text{triplet}}$) These constraints refine advantage ordering among sibling prefixes based on more nuanced observations of their future potential via the current policy's online rollouts. Specifically, for sibling prefixes p_i and p_j that both yield immediate failed online rollouts ($r_i = 0, r_j = 0$) and have not yet exhibited successful completions from the student's online policy ($\neg S(p_i),\ \neg S(p_j)$), we enforce $a_j > a_i$ if p_i is a prefix of a deeper path p_k for which a successful completion *has* been observed ($S(p_k)$). This prioritizes exploration of branches that are currently *less proven* in their overall success trajectory.

$$\mathcal{C}_{\text{triplet}} = \{(i, j, k) \mid \textit{IsSibling}(i, j),\ r_i = 0,\ r_j = 0,\ \neg S(p_i),\ \neg S(p_j),\ \textit{IsPrefix}(i, k),\ S(p_k)\}.$$

B.2 Heuristic Baselines for Advantage Calculation

While the SAE framework is formally defined by the constrained quadratic program in Eq. (2), we explore several practical methods for advantage computation. Our primary approach uses heuristic baselines (e.g., expectation, optimistic) combined with simple mean-centering. This is computationally efficient and preserves the natural scale of prefix-conditioned advantages.[4] For a more direct comparison, we also evaluate an implementation that solves the QP with hard constraints.

For each sampled prefix p_i with reward r_i, we first compute a raw advantage:

$$a'_i = r_i - \alpha\, V(p_i), \quad \alpha \in [0, 1],$$

where $V(p)$ is a heuristic estimate of the expected return from prefix p, summarizing outcomes of rollouts within its subtree. These raw advantages a'_i are then mean-centered to obtain the final advantages a_i, ensuring zero-mean and stabilizing training by reducing bias in gradient estimates.

We consider three heuristic baseline estimators for $V(p)$:

1. **Empirical (Expectation):**
$$V_E(p) = \frac{\#\{\text{successful rollouts from } p\}}{\#\{\text{total rollouts from } p\}},$$
the subtree success rate. By Lemma B.2, the ideal baseline $V^*(p) = \mathbb{E}[r \mid p]$ uniquely minimizes variance and maximizes reward-alignment. $V_E(p)$ is its natural Monte–Carlo approximation. Combined with mean-centering, $a'_i = r_i - \alpha V_E(p_i)$, this yields tree-consistent advantages with near-optimal variance reduction.
2. **Optimistic:**
$$V_O(p) = \mathbf{1}\{\exists \text{ successful rollout in } p\text{'s subtree}\},$$
which amplifies sparse positive signals and encourages exploration.

[4] Normalizing by the standard deviation, as is common in policy-gradient methods, would rescale advantages inconsistently across prefixes and introduce bias into stage comparisons; see App. D.7.

3. **Pessimistic:**

$$V_P(p) = \mathbf{1}\{\nexists \text{ failed rollout in } p\text{'s subtree}\},$$

promoting conservative updates by penalizing uncertainty in downstream paths.

These heuristics capture different assumptions about future trajectories while remaining tree-consistent and computationally light. Together with mean-centering, they approximate the variance- and reward-alignment properties of the ideal solution to (2), offering a practical substitute for exact constrained optimization. Overall, the Empirical (Expectation) baseline tends to deliver the best balance between variance reduction and reward alignment, while the Optimistic and Pessimistic baselines remain valuable in settings such as sparse rewards or high-stakes pruning, where exploration or conservativeness may be more desirable than strict variance minimization.

B.3 Monte Carlo Policy-Gradient Interpretation

Each update samples K prefixes p_k from the MCTS-derived tree. The student policy π_θ then generates full continuations $\hat{c}_k \sim \pi_\theta(\cdot \mid p_k)$ and each completed trajectory yields a binary reward $r_k \in \{0, 1\}$. The policy gradient estimator using staged advantages $A_k = r_k - \alpha V(p_k)$ (Sec. 2) is

$$\widehat{g} = \frac{1}{K} \sum_{k=1}^{K} A_k \, \nabla_\theta \log \pi_\theta(\hat{c}_k \mid p_k),$$

where, for autoregressive models, $\nabla_\theta \log \pi_\theta(\hat{c}_k \mid p_k) = \sum_{t=0}^{|\hat{c}_k|-1} \nabla_\theta \log \pi_\theta(c_{k,t+1} \mid p_k, c_{k,0:t})$.

The ideal stage baseline of Lemma B.2 admits the closed form

$$V(p) = \mathbb{E}[r \mid p] = \Pr_{\pi_\theta}(\text{success} \mid p),$$

i.e., the success probability of continuations from prefix p. We estimate this with Monte–Carlo (MC) rollouts:

$$\widehat{V}_M(p) = \frac{1}{M} \sum_{m=1}^{M} r^{(m)}, \qquad r^{(m)} \in \{0, 1\},$$

(See Appendix D.8 for a short derivation, variance formula and simple concentration bounds.) which is unbiased and satisfies $\mathrm{Var}[\widehat{V}_M(p)] = \mathrm{Var}[r \mid p]/M$. Under sparse rewards this variance can be large, creating a computational tradeoff: reducing estimator variance requires many rollouts.

This tradeoff motivates our practical choices in Sec. 2: mean-centering combined with heuristic baselines (empirical / optimistic / pessimistic) provides a low-cost approximation to $\mathbb{E}[r \mid p]$ while implicitly encoding tree-consistency. By Theorem B.4, the SAE projection of these staged advantages onto the convex set defined by tree-structured order constraints reduces or preserves variance, which directly controls the noise in the MC policy gradient estimator and improves sample efficiency.

B.4 Theoretical Analysis

Key assumptions. We make three short, explicit assumptions used in our statements below:

- (A1) **Constraint consistency.** The ordering set $\mathcal{C}_{\text{order}}$ constructed from MCTS traces is non-contradictory (acyclic). If violated, we use a penalized/soft relaxation.
- (A2) **Convexified/soft normalization.** For theoretical guarantees we analyze a convexified SAE program (RMS equality relaxed to a convex ball) or its penalized counterpart; this leaves the intuitive constraints intact while producing a closed, (locally) well-behaved feasible set .
- (A3) **Policy Gradient Regularity.** The log-policy gradients are uniformly bounded in expectation, a standard condition required to formally analyze the variance of the gradient estimator.

The formal, assumption-aware statements and complete proofs are deferred to Appendix D. Below we state the results used in the text and give short proof sketches to clarify the mechanism. The learning objective is the standard expected policy return, $J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[r(\tau)]$.

Lemma B.1 (Unbiasedness [Wil92])**.** *Let $\widehat{g} = \frac{1}{K} \sum_{k=1}^{K} a_k \, \nabla_\theta \log \pi_\theta(a_k \mid p_k)$, where $a_k = r_k - \alpha V(p_k)$ and $V(p)$ is any deterministic function of the prefix. Then*

$$\mathbb{E}_{p,a \sim \pi_\theta}[\widehat{g}] = \nabla J(\theta).$$

Comment. This standard result confirms that the policy gradient estimate is invariant in expectation to the subtraction of an arbitrary state-dependent baseline $V(p)$. The practical consequence is that the problem of estimator variance can be addressed independently of gradient unbiasedness. This licenses the search for an optimal variance-reducing baseline, which the subsequent lemma provides.

Lemma B.2 (Optimality of Expectation Baseline [GBB04])**.** *The baseline $V(p) = \mathbb{E}[r \mid p]$ minimizes variance among deterministic functions of p and maximizes $\mathrm{Cov}(r, V(p))$; hence it yields the most reward-aligned advantage signal (formal statement and proof in Appendix D.2).*

Comment. The closed-form identity is

$$V(p) = \mathbb{E}[r \mid p] = \Pr_{\pi_\theta}(\text{success} \mid p),$$

so one may estimate it by MC rollouts $\widehat{V}_M(p) = \frac{1}{M}\sum_{m=1}^{M} r^{(m)}$. In sparse-reward domains the variance of this estimator can be large (variance scales as $1/M$), motivating the heuristic baselines (empirical / optimistic / pessimistic) we use in practice; these heuristics trade bias for lower estimator variance while approximately tracking $\mathbb{E}[r \mid p]$ (Sec. 2).

We next make precise how structural (tree) constraints on advantages encode an inductive bias useful for preference updates.

Lemma B.3 (Tree-Induced Advantage Structure and Inductive Bias)**.** *Solving the SAE objective (Eq. 2) or its convex/penalized relaxation with ordering constraints $\mathcal{C}_{\text{order}} = \mathcal{C}_{\text{pair}} \cup \mathcal{C}_{\text{triplet}}$ yields an advantage vector $\boldsymbol{a}^*$ that enforces **prefix-consistent ranking**: for every $(i, j) \in \mathcal{C}_{\text{order}}$ we have $a_i^* + \delta_{ij} \le a_j^*$ (with margin $\delta_{ij} \ge 0$ in the convex program). The precise convexified program and proof are in Appendix D.3.*

Comment. This makes explicit the inductive bias built into SAE: raw rewards are adjusted minimally while obeying the ordering constraints extracted from MCTS traces. The consequence is a set of advantages that respect prefix-consistency, meaning that if one continuation is preferred to another in the tree, the adjusted signal reflects this preference with a margin. The result is not just variance control, but an embedding of structured relational information into the gradient signal itself: triplet constraints re-weight siblings systematically, biasing exploration toward less-proven but promising branches. This inductive bias is the foundation for the guarantees that follow.

Theorem B.4 (Tree Constraints Improve Gradient Signal)**.** *Under the standing assumptions and for a feasible, consistent constraint set, the convex/penalized SAE projection satisfies*

$$\mathrm{Var}[\boldsymbol{a}^*] \le \mathrm{Var}[\boldsymbol{r}],$$

and enforces positive pairwise margins for constrained pairs, thereby improving group-level discriminability used by preference updates. (Precise technical conditions and proof: Appendix D.4.)

Comment. This formalizes the intuition that order-enforcing projections act as a variance-reducing filter on the noisy raw rewards. By projecting onto a closed convex set consistent with ordering, the resulting advantages are guaranteed to have no greater variance than the centered rewards. At the same time, the constraints enforce discriminative margins that improve preference alignment at the group level. This establishes that the structured projection not only stabilizes the signal but also sharpens its usefulness for preference-based updates.

Beyond variance control, SAE also improves the *estimation-to-class* error of prefix values by enforcing tree-consistent structure.

Theorem B.5 (SAE reduces estimation-to-class error)**.** *Let $\widehat{\mathbf{V}}_{\text{GRPO}} \in \mathbb{R}^P$ be the per-prefix empirical means and $\mathcal{C} \subset \mathbb{R}^P$ the closed, convex set of tree-consistent prefix-value vectors. Define $\widehat{\mathbf{V}}_{\text{SAE}} = P_{\mathcal{C}}(\widehat{\mathbf{V}}_{\text{GRPO}})$ and $\mathbf{V}_{\mathcal{C}}^* = P_{\mathcal{C}}(V^\pi)$. Then*

$$\left\|\widehat{\mathbf{V}}_{\text{SAE}} - \mathbf{V}_{\mathcal{C}}^*\right\|_2 \;\le\; \left\|\widehat{\mathbf{V}}_{\text{GRPO}} - \mathbf{V}_{\mathcal{C}}^*\right\|_2,$$

with strict inequality whenever $\widehat{\mathbf{V}}_{\text{GRPO}} \notin \mathcal{C}$ and $P_{\mathcal{C}}(\widehat{\mathbf{V}}_{\text{GRPO}}) \ne P_{\mathcal{C}}(V^\pi)$.

Comment. Projecting per-prefix estimates into the tree-consistent class $\mathcal{C}$ cannot increase—and typically reduces—their distance to the best tree-consistent approximation of the truth. Thus SAE is estimation-optimal *within* the structural class encoded by the ordering constraints.

The appendix contains the formal assumptions, corollaries, and full proofs (Appendix D).

C Extended Experiments

C.1 Comparison of Policy Optimization Strategies

We perform an ablation study comparing several policy optimization strategies for math reasoning tasks, focusing on how supervision signals, whether explicit distillation or task reward, affect final accuracy and resource usage. Table 2 summarizes the reliance of each method on the teacher or critic models, the memory footprint, and the performance. Setup details are in Appendix E.

Method	Teacher	Critic	Memory Cost	Acc. (%)
REINFORCE KL [Aga+24]	✓	✗	High	55.24
REINFORCE KL + baseline [Gu+24]	✓	✓	High	56.72
Actor-Critic (TD-λ) [Gu+24]	✓	✓	High	58.45
MCTS-driven REINFORCE	✓	✗	Medium	67.55
Tree-OPO + REINFORCE KL	✓	(✗)	High	69.90
Tree-OPO (*expectation*)	✗	✗	Low	**77.63**

Table 2: Ablation of policy optimization methods on GSM8K. Compared along supervision requirements, memory usage, and task accuracy.

KL-Distillation Methods. Methods such as `REINFORCE-based KL-distillation` [Aga+24] and `Actor-Critic methods` [Gu+24] supervise the student policy by minimizing (reverse) KL divergence to a teacher. While both distill teacher knowledge, they employ distinct underlying RL frameworks: the former utilizes the REINFORCE algorithm, whereas the latter adopts an Actor-Critic architecture with TD-λ for value estimation. These approaches typically require online student/teacher's rollouts, teacher logits, and critic training (for Actor-Critic), resulting in high memory usage and limited gains (55–58%). Our ablated method, `MCTS-driven REINFORCE`, uses MCTS trajectories to first fine-tune a teacher via supervised learning (SFT), and then applies `GKD` [Aga+24] (`REINFORCE` with reverse KL) to distill that teacher into the student. This avoids expensive teacher rollouts during student training, but still requires on-policy student rollouts, leading to improved efficiency compared to online teacher-guided methods.

Tree-OPO Variants. The final two rows of Table 2 compare variants of our Tree-OPO. The `Tree-OPO + REINFORCE KL` baseline integrates both the task-specific reward and a KL divergence term from a fine-tuned teacher, introducing complexity in balancing multiple learning signals and increasing memory overhead, yet yielding only modest gains (69.90%). Our key finding is that relying solely on task-specific rewards and advantage signals derived from MCTS-structured prefix trees, as in `Tree-OPO`, achieves better accuracy (77.63%) with minimal supervision and resource costs.

Discussion. This ablation highlights the challenge of optimizing hybrid objectives in multi-task settings. While KL-based supervision provides stable gradients, it lacks explicit reward grounding and may underperform in generalization. Tree-OPO avoids this tradeoff by using structure-aware, reward-driven advantage estimation directly over staged reasoning prefixes—resulting in more efficient and effective policy learning.

D Theoretical Proofs

Standing assumptions. We make the following assumptions throughout the appendix (stated explicitly here to keep proofs self-contained):

(A1) **Consistent ordering.** The ordering constraint set $\mathcal{C}_{\text{order}}$ is acyclic (i.e., contains no contradictory pairs).

(A2) **Convexified normalization.** For theoretical statements we replace the RMS equality by the convex ball constraint $\|\mathbf{a}\|^2 \leq N$ (practical solvers may enforce equality or use penalties; see discussion in the main text). This yields a closed, convex and bounded feasible set when combined with linear constraints.

(A3) **Bounded policy gradients.** The policy is such that $\mathbb{E}_{p,a}\left[\|\nabla_\theta \log \pi_\theta(a \mid p)\|^2\right] \leq G^2$.

D.1 Proof of Lemma B.1: Unbiasedness of Gradient Estimate

Lemma D.1 (Unbiasedness of Gradient Estimate [Wil92]). *Let $a \sim \pi_\theta(\cdot \mid p)$ and let $V(p)$ be any deterministic function of the prefix p. Define*

$$\widehat{g} = (r - \alpha V(p))\, \nabla_\theta \log \pi_\theta(a \mid p).$$

Then $\mathbb{E}_{p,a}[\widehat{g}] = \nabla J(\theta)$.

Proof. By the policy gradient theorem with a state-dependent baseline $b(p)$:

$$\nabla J(\theta) = \mathbb{E}_{p,a}\left[(r - b(p))\nabla_\theta \log \pi_\theta(a \mid p)\right].$$

Set $b(p) = \alpha V(p)$. Then

$$\mathbb{E}_{p,a}[\alpha V(p) \nabla_\theta \log \pi_\theta(a \mid p)] = \mathbb{E}_p\Big[\alpha V(p)\, \mathbb{E}_{a\sim\pi_\theta(\cdot|p)}[\nabla_\theta \log \pi_\theta(a \mid p)]\Big].$$

But $\mathbb{E}_{a\sim\pi}[\nabla_\theta \log \pi_\theta(a \mid p)] = \nabla_\theta \sum_a \pi_\theta(a \mid p) = \nabla_\theta 1 = 0$, so the baseline term vanishes. Hence $\mathbb{E}[\widehat{g}] = \nabla J(\theta)$. □

D.2 Proof of Lemma B.2: Optimality of Expectation Baseline

Lemma D.2 (Expectation Minimizes Variance[GBB04]). *Among all deterministic functions $V(p)$, the choice $V(p) = \mathbb{E}[r \mid p]$ (and $\alpha = 1$) minimizes* $\mathrm{Var}[r - \alpha V(p)]$.

Proof. Using the law of total variance for $A = r - \alpha V(p)$:

$$\mathrm{Var}[A] = \mathbb{E}_p[\mathrm{Var}[A \mid p]] + \mathrm{Var}_p[\mathbb{E}[A \mid p]].$$

Since $V(p)$ is deterministic given p, $\mathrm{Var}[A \mid p] = \mathrm{Var}[r \mid p]$, independent of V. Moreover

$$\mathbb{E}[A \mid p] = \mathbb{E}[r \mid p] - \alpha V(p).$$

Thus

$$\mathrm{Var}[A] = \mathbb{E}_p[\mathrm{Var}[r \mid p]] + \mathrm{Var}_p[\mathbb{E}[r \mid p] - \alpha V(p)].$$

The first term is fixed; the second is minimized by setting $\alpha V(p) = \mathbb{E}[r \mid p]$, which yields the stated result. □

D.3 Proof of Lemma B.3: Tree-Induced Advantage Structure

We now formalize the SAE projection used in the main text in a convex form suitable for theoretical guarantees.

Lemma D.3 (Tree-Induced Advantage Structure). *Let $\boldsymbol{r} \in \mathbb{R}^N$ be the observed rewards in a group and let $\mathcal{C}_{\text{order}}$ be an acyclic set of pairwise ordering relations. Consider the convex QP*

$$\begin{aligned} \boldsymbol{a}^* = \arg\min_{\boldsymbol{a}\in\mathbb{R}^N} \ & \|\boldsymbol{a} - \boldsymbol{r}\|^2 \\ \text{s.t.} \quad & \mathbf{1}^\top \boldsymbol{a} = 0, \qquad \|\boldsymbol{a}\|^2 \leq N, \\ & a_i + \delta_{ij} \leq a_j \quad \forall (i,j) \in \mathcal{C}_{\text{order}}, \end{aligned} \tag{2–convex}$$

with fixed margin $\delta_{ij} \geq 0$. Under Assumption (A1) the feasible set is nonempty and convex, the objective is strictly convex, and the problem has a unique solution $\boldsymbol{a}^*$. Moreover $\boldsymbol{a}^*$ satisfies all linear ordering constraints, so that for each $(i,j) \in \mathcal{C}_{\text{order}}$ we have $a_i^* + \delta \leq a_j^*$.

Proof. With linear equalities/inequalities and the convex ball $\|\boldsymbol{a}\|^2 \le N$, the feasible set is convex and closed. The objective $\|\boldsymbol{a} - \boldsymbol{r}\|^2$ is strictly convex (Hessian $2I \succ 0$). A strictly convex function on a nonempty convex compact set has a unique minimizer, so $\boldsymbol{a}^*$ exists and is unique. The ordering constraints are explicit linear constraints in the program, hence the minimizer satisfies them by construction. □

Practical remark. In practice we solve a softened variant (penalty or SLSQP with equality RMS); Appendix E discusses numerical choices. The convex formulation above is used only for the theoretical guarantees (existence, uniqueness, and structural encoding).

D.4 Proof of Theorem B.4: Gradient Signal Improvement

Theorem D.4 (Gradient Signal and Structure)**.** *Under assumptions (A1)–(A3), let $\boldsymbol{r} \in \mathbb{R}^N$ be observed rewards in a group. Define the centered reward vector*

$$\boldsymbol{r}_0 \;=\; \boldsymbol{r} - \overline{\boldsymbol{r}}\,\mathbf{1},$$

and let

$$\mathcal{F}_0 \;=\; \{\, \boldsymbol{a} \in \mathbb{R}^N :\; \mathbf{1}^\top \boldsymbol{a} = 0,\; \|\boldsymbol{a}\|^2 \le N,\; L\boldsymbol{a} \le \mathbf{0} \,\},$$

where the rows of L are $e_i^\top - e_j^\top$ for each $(i,j) \in \mathcal{C}_{\text{order}}$ (zero-margin pairwise orderings). Assume $\mathbf{0} \in \mathcal{F}_0$ (holds under zero margins and acyclicity). Let

$$\boldsymbol{a}^* \;=\; \arg\min_{\boldsymbol{a} \in \mathcal{F}_0} \|\boldsymbol{a} - \boldsymbol{r}_0\|^2$$

be the Euclidean projection of $\boldsymbol{r}_0$ onto $\mathcal{F}_0$. Then:

1. ***(Variance non-increase)** The projection does not increase variance:*

$$\mathrm{Var}[\boldsymbol{a}^*] \;\le\; \mathrm{Var}[\boldsymbol{r}_0] \;=\; \mathrm{Var}[\boldsymbol{r}],$$

where $\mathrm{Var}[\boldsymbol{x}] = \frac{1}{N}\|\boldsymbol{x}\|^2$ for zero-mean vectors.

Moreover, equality holds if and only if $\boldsymbol{r}_0 \in \mathcal{F}_0$; otherwise the inequality is strict.

Variance non-increase via convex projection. By construction $\boldsymbol{a}^*$ is the Euclidean projection of $\boldsymbol{r}_0$ onto the closed convex set $\mathcal{F}_0$. Projection optimality (firm nonexpansiveness / characterization of the projector; see [BC17, Thm. 3.14]) gives the variational inequality

$$\langle \boldsymbol{r}_0 - \boldsymbol{a}^*,\, \boldsymbol{z} - \boldsymbol{a}^* \rangle \le 0 \qquad \forall \boldsymbol{z} \in \mathcal{F}_0.$$

Since $\mathbf{0} \in \mathcal{F}_0$ we may take $\boldsymbol{z} = \mathbf{0}$, yielding

$$\langle \boldsymbol{r}_0 - \boldsymbol{a}^*,\, -\boldsymbol{a}^* \rangle \le 0 \quad\Longrightarrow\quad \langle \boldsymbol{r}_0 - \boldsymbol{a}^*,\, \boldsymbol{a}^* \rangle \ge 0.$$

Hence

$$\|\boldsymbol{a}^*\|^2 \le \langle \boldsymbol{r}_0, \boldsymbol{a}^* \rangle \le \|\boldsymbol{r}_0\|\,\|\boldsymbol{a}^*\|,$$

and dividing by $\|\boldsymbol{a}^*\|$ (if $\boldsymbol{a}^* \neq \mathbf{0}$) gives $\|\boldsymbol{a}^*\| \le \|\boldsymbol{r}_0\|$. If $\boldsymbol{a}^* = \mathbf{0}$ the inequality is trivial. Because both $\boldsymbol{a}^*$ and $\boldsymbol{r}_0$ are zero-mean, their population variances satisfy

$$\mathrm{Var}[\boldsymbol{a}^*] \;=\; \frac{1}{N}\|\boldsymbol{a}^*\|^2 \;\le\; \frac{1}{N}\|\boldsymbol{r}_0\|^2 \;=\; \mathrm{Var}[\boldsymbol{r}_0].$$

This establishes the variance non-increase property. The inequality is strict exactly when $\boldsymbol{r}_0 \notin \mathcal{F}_0$ (otherwise the projection returns $\boldsymbol{r}_0$ and norms are equal). Thus whenever the centered rewards violate at least one tree constraint the projection strictly reduces squared norm (hence variance).

Role of the ordering constraints. The linear order constraints $L\boldsymbol{a} \le 0$ encode *tree consistency* by enforcing prefix-relative advantage orderings. With the convexified normalization $\|\boldsymbol{a}\|^2 \le N$, the feasible set

$$\mathcal{F}_0 = \big\{\boldsymbol{a} : \mathbf{1}^\top \boldsymbol{a} = 0,\;\; \|\boldsymbol{a}\|^2 \le N,\;\; L\boldsymbol{a} \le 0\big\}$$

is closed, convex, and contains $\mathbf{0}$. Therefore, the Euclidean projection

$$\boldsymbol{a}^* = \mathrm{Proj}_{\mathcal{F}_0}(\boldsymbol{r}_0)$$

satisfies $\|\boldsymbol{a}^*\| \le \|\boldsymbol{r}_0\|$ and hence $\mathrm{Var}[\boldsymbol{a}^*] \le \mathrm{Var}[\boldsymbol{r}]$. If the unconstrained (GRPO) projection already respects the orderings, the two projections coincide. Otherwise, $\boldsymbol{a}^*$ enforces the tree structure by making the minimal modification (in Euclidean norm) to $\boldsymbol{r}_0$ necessary to satisfy the order constraints. In practice, this structural correction typically—and often strictly—reduces the advantage variance across prefixes when $\boldsymbol{r}_0$ violates tree constraints. Note, however, that there is no universal monotonicity guarantee comparing $\|\mathrm{Proj}_{\mathcal{F}_0}(\boldsymbol{r}_0)\|$ with $\|\mathrm{Proj}_H(\boldsymbol{r}_0)\|$ for every possible $\boldsymbol{r}_0$ (projections onto nested convex sets need not be ordered). Consequently, we present the projection variance bound as the rigorous core result and treat the additional empirical benefit relative to plain GRPO as an expected, often-observed effect supported by experiments. □

D.5 GRPO as a Degenerate Projection and Variance Comparison

Corollary D.5 (SAE does not increase variance; strict reduction under violations)**.** *Let $\mathbf{r} \in \mathbb{R}^N$ be observed rewards and $\mathbf{r}_0 = \mathbf{r} - \bar{r}\mathbf{1}$ the centered rewards. Assume $\mathcal{F} \subseteq H := \{\mathbf{a} : \mathbf{1}^\top \mathbf{a} = 0\}$ is closed, convex and satisfies $\mathbf{0} \in \mathcal{F}$ and $\|\mathbf{a}\|^2 \le N$ for all $\mathbf{a} \in \mathcal{F}$. Denote $\mathbf{a}^* = \Pi_{\mathcal{F}}(\mathbf{r}_0)$ (SAE projection) and $\mathbf{a}_{\mathrm{GRPO}} = \mathbf{r}_0$ (plain mean-centering). If $\mathbf{r}_0 \neq \mathbf{0}$ define the std-normalized GRPO vector $\mathbf{a}_{\mathrm{GRPO\_std}} = \sqrt{N}\,\mathbf{r}_0/\|\mathbf{r}_0\|$. Then:*

1. *$\mathrm{Var}[\mathbf{a}^*] \le \mathrm{Var}[\mathbf{r}_0] = \mathrm{Var}[\mathbf{a}_{\mathrm{GRPO}}]$, with strict inequality whenever $\mathbf{r}_0 \notin \mathcal{F}$.*
2. *For the nondegenerate case $\mathbf{r}_0 \neq \mathbf{0}$, $\mathrm{Var}[\mathbf{a}^*] \le 1 = \mathrm{Var}[\mathbf{a}_{\mathrm{GRPO\_std}}]$; hence SAE never increases variance relative to GRPO with std-normalization.*

In particular, when centered rewards violate tree-order constraints the SAE projection typically (and often strictly) reduces variance compared to plain mean-centering.

Proof. (1) By definition $\mathbf{a}^*$ is the Euclidean projection of $\mathbf{r}_0$ onto the closed convex set $\mathcal{F}$. Projection optimality yields, for all $\mathbf{z} \in \mathcal{F}$, $\langle \mathbf{r}_0 - \mathbf{a}^*,\ \mathbf{z} - \mathbf{a}^* \rangle \le 0$. Taking $\mathbf{z} = \mathbf{0} \in \mathcal{F}$ gives $\langle \mathbf{r}_0 - \mathbf{a}^*,\ \mathbf{a}^* \rangle \ge 0$. Expanding norms:

$$\|\mathbf{r}_0\|^2 = \|\mathbf{r}_0 - \mathbf{a}^*\|^2 + 2\langle \mathbf{r}_0 - \mathbf{a}^*, \mathbf{a}^* \rangle + \|\mathbf{a}^*\|^2 \ge \|\mathbf{a}^*\|^2,$$

hence $\|\mathbf{a}^*\|^2 \le \|\mathbf{r}_0\|^2$. Dividing by N gives $\mathrm{Var}[\mathbf{a}^*] \le \mathrm{Var}[\mathbf{r}_0]$. If $\mathbf{r}_0 \notin \mathcal{F}$ then $\|\mathbf{r}_0 - \mathbf{a}^*\| > 0$ and the inequality is strict.

(2) For $\mathbf{r}_0 \neq \mathbf{0}$ the std-normalized GRPO vector has $\|\mathbf{a}_{\mathrm{GRPO\_std}}\|^2 = N$, so $\mathrm{Var}[\mathbf{a}_{\mathrm{GRPO\_std}}] = 1$. Since by assumption every $\mathbf{a} \in \mathcal{F}$ satisfies $\|\mathbf{a}\|^2 \le N$, we have $\|\mathbf{a}^*\|^2 \le N$, hence $\mathrm{Var}[\mathbf{a}^*] \le 1 = \mathrm{Var}[\mathbf{a}_{\mathrm{GRPO\_std}}]$. This proves that SAE does not increase variance compared to GRPO with std-normalization. □

D.6 SAE Improves Estimation of the Best Consistent Value Function

Recall from Lemma B.2 that the variance-minimizing deterministic baseline for a prefix is the true prefix-value

$$V^\pi(p) \;=\; \mathbb{E}_{\tau\sim\pi}\big[r(\tau) \mid \tau \supset p\big],$$

i.e. the expected terminal reward of trajectories that extend prefix p.

We consider three concrete empirical/algorithmic estimators of V^π:

$$\textbf{(GRPO, global)} \qquad V_{\mathrm{GRPO}} \;=\; \bar{r} \;=\; \frac{1}{N}\sum_{i=1}^{N} r(\tau_i), \tag{3}$$

$$\textbf{(Subtree empirical / } V_E\textbf{)} \qquad V_E(p) \;=\; \frac{1}{N_p^{\mathrm{sub}}} \sum_{\tau_i : \tau_i \supset p} r(\tau_i), \tag{4}$$

$$\textbf{(SAE / projected)} \qquad \widehat{\mathbf{V}}_{\mathrm{SAE}} \;=\; P_{\mathcal{C}}\big(\widehat{\mathbf{V}}_{\mathrm{GRPO}}\big), \qquad \widehat{V}_{\mathrm{SAE}}(p) \;=\; \big(\widehat{\mathbf{V}}_{\mathrm{SAE}}\big)_p. \tag{5}$$

Here $\{\tau_i\}_{i=1}^N$ are the sampled completions, N_p^{sub} counts all sampled completions that extend prefix p, and $\widehat{\mathbf{V}}_{\mathrm{GRPO}} \in \mathbb{R}^P$ is the vector of per-prefix empirical exact-start means $\widehat{V}_{\mathrm{GRPO}}(p) =$

$\frac{1}{n_p^{\text{exact}}} \sum_{i:s_i=p} r(\tau_i)$ (used as the input to projection). The set $\mathcal{C} \subset \mathbb{R}^P$ denotes the closed convex class of prefix-value vectors satisfying the tree-ordering constraints (convexified as needed).

Equation (5) is the *idealized* prefix-space SAE: it projects the empirical per-prefix vector onto $\mathcal{C}$. In practice we realize SAE by a sample-space projection (solve the QP in sample-space for advantages, then aggregate), which is equivalent to the prefix-space projection under Assumption A4 (constraints act only on prefix-aggregates); otherwise the two may differ slightly due to solver/regularization details.

Theorem D.6 (SAE optimally reduces estimation-to-class error)**.** *Let $\mathbf{V}_{\mathcal{C}}^* = P_{\mathcal{C}}(\mathbf{V}^\pi)$ be the projection of the true prefix-value vector onto $\mathcal{C}$. Then for any empirical GRPO prefix-vector $\widehat{\mathbf{V}}_{\text{GRPO}}$,*

$$\left\|\widehat{\mathbf{V}}_{\text{SAE}} - \mathbf{V}_{\mathcal{C}}^*\right\|_2 \;\le\; \left\|\widehat{\mathbf{V}}_{\text{GRPO}} - \mathbf{V}_{\mathcal{C}}^*\right\|_2,$$

with strict inequality whenever $\widehat{\mathbf{V}}_{\text{GRPO}} \notin \mathcal{C}$ and $P_{\mathcal{C}}(\widehat{\mathbf{V}}_{\text{GRPO}}) \neq P_{\mathcal{C}}(\mathbf{V}^\pi)$.

Sketch. Immediate from the distance-decreasing property of Euclidean projection onto a closed convex set: for every $z \in \mathcal{C}$ and any $x \in \mathbb{R}^P$ we have $\|P_{\mathcal{C}}(x) - z\|_2 \le \|x - z\|_2$. Set $x = \widehat{\mathbf{V}}_{\text{GRPO}}$ and $z = \mathbf{V}_{\mathcal{C}}^*$. □

Comparison and limitations.

- V_E **vs** V_{GRPO}**.** The subtree estimator $V_E(p)$ (Eq. 4) trades variance for potential bias at each prefix. The MSE of an estimator is decomposed into its squared bias and variance: $\text{MSE}(\widehat{V}) = \text{Bias}(\widehat{V})^2 + \text{Var}(\widehat{V})$. While V_{GRPO} is an unbiased estimator of $V^\pi(p)$ (zero bias, high variance), $V_E(p)$ is a biased estimator (non-zero bias, low variance) due to pooling samples from different states. A good V_E estimator achieves a larger reduction in variance than the square of its introduced bias.
- V_{SAE} **vs** V_{GRPO}**.** Theorem D.6 guarantees that projecting GRPO estimates into $\mathcal{C}$ reduces the estimation-to-class distance in ℓ_2. This is an *estimation* improvement relative to the feasible class $\mathcal{C}$, but it does not by itself imply $\widehat{V}_{\text{SAE}}$ is strictly closer to the true V^π than $\widehat{V}_{\text{GRPO}}$ unless $\mathbf{V}^\pi$ is well-approximated by $\mathcal{C}$.
- V_E **vs** V_{SAE}**.** There is no unconditional ordering: V_E trades variance for (potential) bias, while V_{SAE} enforces structural consistency and reduces estimation-to-class error. Which estimator is closer to V^π depends on (i) how well $\mathcal{C}$ matches $\mathbf{V}^\pi$ and (ii) the bias magnitude of V_E within each subtree.

D.7 On mean-centering without standardization

A common practice in policy-gradient methods is to normalize advantages by both mean and standard deviation (z-scoring). However, in the staged setting this can distort reward magnitudes and interfere with the tree-structured consistency of prefix values. Our approach applies only mean-centering, which is both simpler and theoretically justified.

Claim D.7 (Mean-centering is the variance-optimal shift; scaling is conventional). Let $\{A_k\}_{k=1}^K$ be advantage estimates with finite variance. Consider affine transforms $A'_k = (A_k - b)/c$ with $c > 0$ and the stochastic policy-gradient estimator

$$\widehat{g} \;=\; \frac{1}{K} \sum_{k=1}^{K} A'_k \, \nabla_\theta \log \pi_\theta(a_k \mid s_k).$$

For any fixed $c > 0$, the choice $b = \mathbb{E}[A]$ minimizes the (upper-bounded) variance of $\widehat{g}$. Moreover, changing c simply rescales $\widehat{g}$ (and its variance) by $1/c$ (resp. $1/c^2$) and can be absorbed into the learning rate. Hence we set $c = 1$ to preserve reward scale and avoid distorting tree-relative magnitudes.

Proof. Using the standard bound on score-function moments (Assumption (A3)),

$$\text{Var}(\widehat{g}) \;=\; \text{Var}\Big(\frac{1}{K}\sum_{k=1}^{K} \tfrac{A_k - b}{c} \, \nabla_\theta \log \pi_\theta(a_k \mid s_k)\Big) \;\le\; \frac{1}{K}\,\frac{G^2}{c^2}\,\text{Var}(A - b).$$

For fixed $c > 0$, the right-hand side is minimized over b by $b = \mathbb{E}[A]$, since $\mathrm{Var}(A - b)$ is minimized by mean-centering. The factor $1/c^2$ shows that c only rescales the estimator and its variance; in practice c trades off identically with the learning rate and does not improve the signal beyond a step-size change. We therefore adopt $c = 1$ by convention to retain the native reward scale. □

This explains why simple mean-centering suffices: it provides the variance-optimal *shift* while avoiding the instability and semantic distortion introduced by standardization of the *scale*.[5]

D.8 MC baseline: unbiasedness and variance

Let $r^{(1)}, \ldots, r^{(M)} \in \{0, 1\}$ be independent outcomes of rollouts from prefix p under π_θ. Define the empirical estimator

$$\widehat{V}_M(p) = \frac{1}{M} \sum_{m=1}^{M} r^{(m)}.$$

Unbiasedness is immediate:

$$\mathbb{E}[\widehat{V}_M(p)] = \mathbb{E}[r \mid p] = V(p).$$

The variance satisfies

$$\mathrm{Var}[\widehat{V}_M(p)] = \frac{\mathrm{Var}[r \mid p]}{M} = \frac{V(p)(1 - V(p))}{M},$$

since $r \in \{0, 1\}$. By the Central Limit Theorem, for large M,

$$\widehat{V}_M(p) \approx \mathcal{N}\big(V(p),\, V(p)(1 - V(p))/M\big),$$

so a (Gaussian) approximate $1 - \delta$ confidence interval has half-width $\Phi^{-1}(1 - \delta/2)\sqrt{V(p)(1 - V(p))/M}$. A conservative non-asymptotic bound follows from Hoeffding's inequality:

$$\Pr\left(|\widehat{V}_M(p) - V(p)| \geq \epsilon\right) \leq 2\exp(-2M\epsilon^2).$$

These identities quantify the compute–variance tradeoff: sparse rewards (small $V(p)$) require large M to reduce baseline noise, motivating the low-cost, biased heuristics we use in practice (Sec. 2).

E Experimental Setup: Hyperparameters and Model Configuration

This section details the hyperparameter and model settings used for both Tree-OPO and GPRO training as well as the training setting combines the distillation-based methods.

Policy Optimization Objectives. Our policy π_θ is optimized using a combined objective, L_{policy}, which integrates a GRPO-style policy gradient term, *optionally* with a distillation loss:

$$L_{\text{policy}} = -\mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \log \pi_\theta(a_t|s_t) A^{\pi_\theta}(s_t, a_t) \right] + \beta_{\text{distill}} D_{\text{KL}}(\pi_{\text{teacher}}(a_t|s_t) \| \pi_\theta(a_t|s_t))$$

The first term represents the policy gradient loss (negative of the objective), where $A^{\pi_\theta}(s_t, a_t)$ is the advantage from Section 2. The α parameter (0.5) serves as the weighting coefficient for the baseline component within the advantage estimation ($a'_i = r_i - \alpha V(p_i)$). The second term is an optional distillation loss. When distillation is applied, it is weighed by the `distillation coefficient` ($\beta_{\text{distill}} = 0.1$). Otherwise, β_{distill} is set to 0. We utilize **reverse KL divergence**, $D_{\text{KL}}(P\|Q)$, where P is the distribution of teacher policies (π_{teacher}) and Q is the distribution of student policies (π_θ). This choice promotes student coverage of the teacher's distributional modes, crucial for tasks with multiple valid reasoning paths. Separate learning rates are specified for the primary GRPO policy (`3e-5`) and for the distillation loss component (`5e-4`), or in specific distillation-only training regimes *when distillation is enabled*.

[5] Z-scoring may aid cross-task comparability, but in tree-structured reasoning the absolute scale carries information about prefix difficulty and should be preserved.

Parameter	Value
α (Advantage Baseline Weight)	0.5
Distillation Coefficient (β_{distill})	0.1
Model Precision	bf16
Teacher Precision	bf16
Group Size	8
Max Prompt Length	256 (GRPO), 512 (Tree-OPO)
Max Sequence Length	512 (GRPO), 768 (Tree-OPO)
Top-k (Sampling)	0
Top-p (Sampling)	1.0
Temperature (Sampling)	1.0
KL Divergence Type	Reverse
Optimizer	AdamW
Batch size	32
Learning rate (Policy, GRPO)	3e-5
Learning rate (Policy, Distillation)	5e-4
LR Scheduler	Cosine
Critic Head Architecture	Llama Decoder
Critic Warm-up Epochs	20
Critic Algorithm	Actor-Critic with TD(λ)
GAE_gamma (γ)	0.95
PEFT LoRA Rank (r)	16
PEFT LoRA Alpha (α_{LoRA})	64
PEFT LoRA Dropout	0.1

Table 3: Hyperparameter and model settings for Tree-PRO, encompassing GRPO and distillation components.

Critic Model and Loss. The critic network learns a state-value function, $V_\phi(s)$, to estimate expected returns from prefixes. Its `Llama Decoder` architecture allows it to process sequence contexts for value prediction `GAE_gamma` ($\gamma = 0.95$) is the discount factor for Generalized Advantage Estimation (GAE). The critic's loss, L_{critic}, is a Mean Squared Error (MSE) between its prediction and the computed target value:

$$L_{\text{critic}} = \frac{1}{N} \sum_{i=1}^{N} (V_\phi(s_i) - V_i^{\text{target}})^2$$

where V_i^{target} is the Generalized Advantage Estimation (GAE) return for state s_i.

Additional Model Settings. All models, including student and teacher, utilize `bf16` precision. Generation sampling employs `Top-p` ($p = 1.0$) with `Temperature` 1.0. The `Group Size` for GRPO batching is 8. Standard GRPO configurations use maximum prompt and sequence lengths of 256 and 512 tokens respectively, while Tree-OPO setups extend these to 512 and 768. Parameter Efficient Fine-Tuning (PEFT) via LoRA is applied, configured with a LoRA `rank` of 16, a scaling `alpha` of 64, and a `dropout` rate of 0.1.

SAE Formulations and Solver. We implement SAE by solving the constrained optimization problem in Eq. 2. We consider two primary formulations:

The **SAE (*hard*)** version, evaluated in our experiments, enforces a strict margin of $\delta = 0.01$ on triplet-consistency constraints alongside the non-convex equality normalization $\|\boldsymbol{a}\|^2 = N$.

The **SAE (*soft*)** version is a convex relaxation used for our theoretical analysis. This formulation uses the inequality $\|\boldsymbol{a}\|^2 \leq N$ and typically zero margins to guarantee a unique, well-behaved solution. An alternative soft approach converts the ordering constraints into a penalty term in the objective:

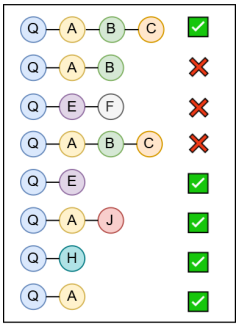
$$\min_{\boldsymbol{a} \in \mathbb{R}^N} \quad \|\boldsymbol{a} - \boldsymbol{r}\|^2 + \lambda \sum_{(i,j) \in \mathcal{C}_{\text{order}}} \max(0, a_i - a_j + \delta)^2.$$

Our current results are based on the *hard*-constrained formulation; a full evaluation of the penalized soft version is deferred to future work. Both formulations can be solved using standard numerical

methods; we use `scipy.optimize.minimize` with the SLSQP algorithm, warm-started from mean-centered rewards.

F Staged Advantage Estimation: A Minimal Example

To illustrate the structure of our staged prompting setup and the computation of advantages, we present a minimal example derived from the MCTS reasoning tree. Each prompt chain represents an incremental sequence of reasoning steps towards a solution, beginning with a base question (Q) and extended by alphabet-labeled stages (e.g., A, B, C).



(a) A group of staged prompts.

Node	Text
Q	A basketball team won 40 games in their last competition. In the next competition, they won 5/8 times as many games as they won in their first competition. In the third competition, they won the same number of games as the sum of the first and second competition winnings. Calculate the total number of wins they had in the three competitions.
A	They won 5/8 times as many games as they won in their first competition, which is 40 * 5/8 = 25 games. The answer is 25.
B	They won the same number of games as the sum of the first and second competition winnings, which is 40 + 25 = 65 games. The answer is 65.
C	To calculate the total number of wins, add up all the wins: 40 (first competition) + 25 (second competition) + 65 (third competition) = 130.
E	The team won 5/8 times as many games as they won in their first competition, which is (5/8) * 40 = 25 games. The answer is 25.
F	To get the number of games the team won in the third competition, add the number of games won in the first two competitions.
J	Multiply the number of games won in the first competition by 5/8 to determine the number of games won in the second competition.
H	Determine the number of games won in the second competition. It was stated that the team won 5/8 times as many games in the second competition as they did in the first. Since they won 40 games in the first competition, we calculate \\(\\frac{5}{8} \\times 40 \\).

(b) Correspondence between reasoning nodes and textual content. Each node represents a reasoning step in a multi-hop prompt.

Figure 4: **Illustration of Staged Reasoning Prompts and Text Mappings.** (a) A group of example staged prompts. Each row is a distinct prompt chain, and the checkmark/cross indicates whether the model produced a correct final answer when conditioned on that prompt. (b) Correspondence between symbolic reasoning nodes (Q, A, B, etc.) and their textual content. Each node represents a distinct reasoning step, forming compositional instructions separated by double newlines (`\n\n`) in the actual prompt fed to the language model.

Table 4 presents success statistics and heuristic value estimations ($V_{\text{exp}}(x)$, $V_{\text{opt}}(x)$, $V_{\text{pes}}(x)$). As detailed in Section 2, advantage optimization is guided by **tree consistency constraints** ($\mathcal{C}_{\text{order}}$), ensuring coherent advantage signals.

The **expected value baseline** ($V_{\text{exp}}(x)$), calculated as average success rate, inherently aligns with these constraints, yielding precise 'surprise' signals. For instance, '$\mathcal{C}_{\text{pair}}$' requires $a_i < a_j$ for a failing parent p_i leading to a successful child p_j. Comparing 'Q-A-B' ($E[R] = 1/3$, $r = 0$) and its successful descendant 'Q-A-B-C' ($E[R] = 1/2$, $r = 1$ from 't0'), V_{exp} generates advantages of $-1/6$ and $3/4$ respectively, satisfying $a_i < a_j$. While this example doesn't fully capture all '$\mathcal{C}_{\text{triplet}}$' conditions, V_{exp}'s accurate value estimations ($E[R|\text{Q-A-B}] = 1/3$ vs $E[R|\text{Q-E-F}] = 0$) are fundamental for enabling such fine-grained control over advantage ordering.

Conversely, **optimistic** ($V_{\text{opt}}(x)$) and **pessimistic** ($V_{\text{pes}}(x)$) baselines often violate these constraints. $V_{\text{pes}}(\text{Q-E-F}) = 0$ and $V_{\text{pes}}(\text{Q-A-B}) = 0$ despite different $E[R]$. Similarly, V_{opt} assigns 1 to 'Q-A-B', 'Q-A-B-C', and 'Q-A-J' (all having different $E[R]$), losing vital discriminative power for tree consistency.

Table 5 demonstrates staged advantage computation. The **expectation baseline** (V_{exp}) best satisfies the *tree consistency constraints*. Its accurate reflection of expected rewards ensures advantages consistently signal better/worse-than-expected outcomes relative to a path's true potential. This yields informative, context-sensitive advantages. In contrast, V_{opt} and V_{pes}'s coarse assignments lead to misleading signals, failing tree consistency.

G Multi-Turn MCTS Rollout Structure and Tree Visualization

Multi-turn Rollout. Following rStar [Qi+24], a reasoning episode is decomposed into stages $t = 1, \ldots, T$, where at each stage the agent generates an action a_t conditioned on a state s_t, and transitions to the next state via $s_{t+1} = s_t \circ a_t$. This defines a multi-turn rollout, where the final

x / Prompt	Success	Total	$V_{\text{exp}}(x)$	$V_{\text{opt}}(x)$	$V_{\text{pes}}(x)$
Q-A	3	5	$\frac{3}{5}$	1	0
Q-A-B	1	3	$\frac{1}{3}$	1	0
Q-A-B-C	1	2	$\frac{1}{2}$	1	0
Q-A-J	1	1	1	1	0
Q-E	1	2	$\frac{1}{2}$	1	0
Q-E-F	0	1	0	0	0
Q-H	1	1	1	1	1

Table 4: Success statistics and corresponding heuristic value baselines for example reasoning prefixes. $V_{\text{exp}}(x)$, $V_{\text{opt}}(x)$, and $V_{\text{pes}}(x)$ are derived from observed success rates within the prefix's subtree, reflecting different assumptions of future success.

R	$R - \frac{1}{2}V_{\text{exp}}(x)$	$R - \frac{1}{2}V_{\text{opt}}(x)$	$R - \frac{1}{2}V_{\text{pes}}(x)$
1	$\frac{7}{10}$	$\frac{1}{2}$	1
0	$-\frac{1}{6}$	$-\frac{1}{2}$	0
0	0	$-\frac{1}{2}$	0
0	$-\frac{1}{4}$	$-\frac{1}{2}$	0
1	$\frac{3}{4}$	$\frac{1}{2}$	1
1	$\frac{1}{2}$	$\frac{1}{2}$	1
1	$\frac{1}{2}$	1	1
1	$\frac{3}{5}$	$\frac{1}{2}$	$\frac{1}{2}$

Table 5: Example staged advantage values computed as $R - \alpha V(x)$ for different heuristic baselines. The **expectation baseline** (V_{exp}; second column, shaded) better satisfies the *tree-consistency constraints*, reflecting coherent advantage values aligned with reward structure across the tree. In contrast, the **optimistic** (V_{opt}) and **pessimistic** (V_{pes}) baselines exhibit more inconsistent or overly conservative adjustments. Each row corresponds to a prompt-completion trajectory with binary reward R from Figure 4(a).

solution is constructed as a chain of reasoning steps $\{a_1, \ldots, a_T\}$. Unlike standard single-step MCTS, which performs search from a fixed root to a terminal state, rStar applies MCTS recursively over multiple stages, building up partial completions and branching decisions at each intermediate node. The result is a structured, non-Markovian tree of reasoning trajectories, where each visited prefix s_t corresponds to a meaningful subproblem along a candidate path. Thus, the training data is not drawn directly from an offline teacher policy, but rather sampled from a *complex, multi-turn distribution* induced by search—reflecting the teacher's internal planning dynamics across many possible intermediate reasoning paths. Figure 5 iillustrates a reasoning tree produced by the multi-turn MCTS rollout process.

Learning from the MCTS-Induced Distribution. Although the MCTS-derived distribution over prefixes does not correspond to samples from a stationary expert policy, it is shaped by a structured search process guided by value estimates and exploration. Each prefix s_t encountered during multi-turn rollout reflects a plausible intermediate reasoning state prioritized during search. From an RL perspective, this yields an offline dataset composed of staged pairs (s_t, c_t, R_t), where completions c_t and rewards R_t are conditioned on prefix context. This structured distribution captures diverse yet high-quality trajectories that extend beyond single-path demonstrations. By leveraging this data with policy gradient methods and advantage estimation, the student can effectively optimize its policy without requiring direct imitation. Thus, the learning process exploits the inductive bias of the MCTS-induced distribution to supervise training in a way that aligns with structured reasoning dynamics.

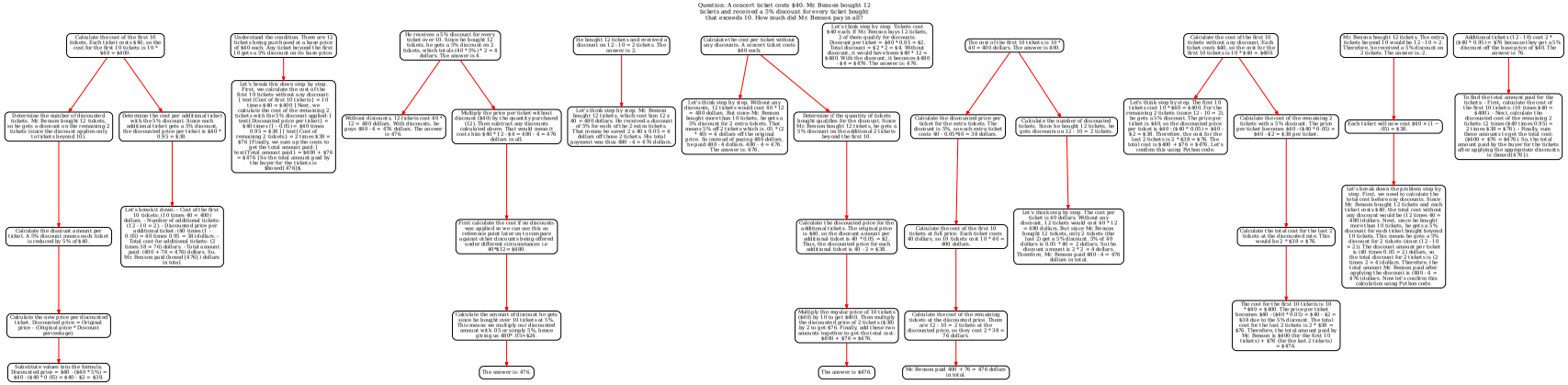


Figure 5: A visualization of a multi-turn MCTS tree to a question in GSM8k.

H Algorithm

Input: Teacher MCTS budget $(B, d_{\max}, b)$; student policy π_θ; batch size K; baseline rollouts M; scaling α; solver option; learning rate η
Output: Updated parameters θ
Offline MCTS:
foreach *training problem* **do**
  Run multi-turn MCTS (B rollouts, depth $\leq d_{\max}$, branching $\leq b$)
  Collect trajectories and augment with all prefixes $\rightarrow$ prefix tree $\mathcal{T}$
end
while *not converged* **do**
  Sample K prefixes $\{p_k\} \subset \mathcal{T}$
  for $k = 1, \ldots, K$ **do**
    Rollout $\hat{c}_k \sim \pi_\theta(\cdot \mid p_k)$; reward $r_k \in \{0, 1\}$
  end
  Estimate baseline $V(p_k)$ (*MC*, empirical, optimistic/pessimistic)
  Compute $a'_k \leftarrow r_k - \alpha V(p_k)$; mean-center $\{a'_k\} \rightarrow \{a_k\}$
  if *SAE enabled* **then**
    Refine $\{a_k\}$ via heuristic or constrained QP
  end
  $\widehat{g} \leftarrow \frac{1}{K} \sum_k a_k \nabla_\theta \log \pi_\theta(\hat{c}_k \mid p_k)$
  Update $\theta \leftarrow \theta + \eta\, \widehat{g}$
end

Algorithm 1: Tree-OPO: Tree-structured Relative Policy Optimization