

Heterogeneous Agent Collaborative Reinforcement Learning

Zhixia Zhang^{1*} Zixuan Huang^{1,2*} Xin Xia² Deqing Wang¹ Fuzhen Zhuang¹ Shuai Ma¹ Ning Ding³
Yaodong Yang⁴ Jianxin Li¹ Yikun Ban¹

Abstract

In scenarios where multiple agents are to be optimized for the same task, vanilla Reinforcement Learning (RL) paradigms constrain each agent to execute tasks independently and update based solely on its own rollouts, they repeatedly generate trajectories and yield verifiable rewards, while these costly intermediate results are only utilized for self-training. To break through this wasteful practice, we introduce **HACPO (Heterogeneous Agent Collaborative Policy Optimization)** to achieve this goal. It is a novel collaborative multi-agent RL training paradigm that enables rollout sharing, maximizing rollout utilization and facilitating mutual knowledge transfer among heterogeneous agents. To effectively bridge the discrepancies in capability and policy distribution, we introduce four tailored modifications and rigorously prove the unbiasedness of these improvements in advantage estimation and the validity of the optimization direction through theoretical analysis. Experimental results demonstrate that HACPO consistently achieves superior accuracy across various heterogeneous model combinations and datasets. This work points to a promising new direction for efficient collaborative learning among multi-agent systems.

1. Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as a highly effective paradigm for training strong reasoning models via automatically checkable reward signals (e.g., unit tests and formal verifiers)(Yang et al., 2026b). Compared with SFT (Zou et al., 2025; Chen et al., 2025; Ouyang et al., 2022a) and DPO (Rafailov et al., 2023; Huang et al., 2025), RL Optimization (Stiennon et al., 2020)

For implement details, feel free to contact Zhixia Zhang 22376220@buaa.edu.cn. * Equal Contribution. ¹Beihang University ²Bytedance China ³Tsinghua University ⁴Peking University. Correspondence to: Yikun Ban <yikunb@buaa.edu.cn>.

Preprint. February 20, 2026.

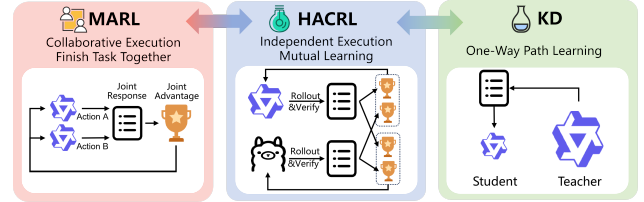


Figure 1. The significant differences among Multi-Agent RL, Knowledge Distillation, and the proposed HACPO. HACPO targets independent execution with collaborative optimization.

more directly aligns the model with downstream objectives, and RLVR further strengthens this alignment through verifiability. Within RLVR, group-based policy optimization algorithms such as GRPO (Shao et al., 2024; Yang et al., 2026b) replace the critic in PPO (Schulman et al., 2017) by computing group-relative advantages (Yang et al., 2026b), motivating variants including DAPO (Yu et al., 2025) and GSPO (Zheng et al., 2025). Despite these advances, RLVR remains bottlenecked by expensive on-policy sampling and verification, which frequently dominate the overall training overhead and limit scalability. Meanwhile, modern LLM ecosystems are inherently *heterogeneous*: agents differ in parameter states, model size, architecture, and are often designed or adapted for different downstream tasks, such as instruction following (Ouyang et al., 2022b), mathematical problem solving (Cobbe et al., 2021), and code generation (Weyssow et al., 2025). This heterogeneity becomes even more pronounced when models come from different vendors or families (Yang et al., 2025a; Grattafiori et al., 2024), with mismatched pretraining corpora, tokenizers, and architectural choices.

Typically, given *one* identical task, *multiple* agents execute RLVR optimization *independently* of one another. For essentially the same objective, they repeatedly generate trajectories and yield verifiable rewards, while these costly intermediate results are only utilized for self-training.

To break through this wasteful practice, we propose a collaborative policy optimization problem for RLVR: *given a set of heterogeneous agents, can an agent improve both effectiveness and efficiency by leveraging rollouts generated by other agents, rather than relying solely on its own on-policy rollouts?* Our goal is to enable *mutual benefit* across agents—each agent can reuse rollouts from others—while controlling distribution shift induced by heterogeneity.

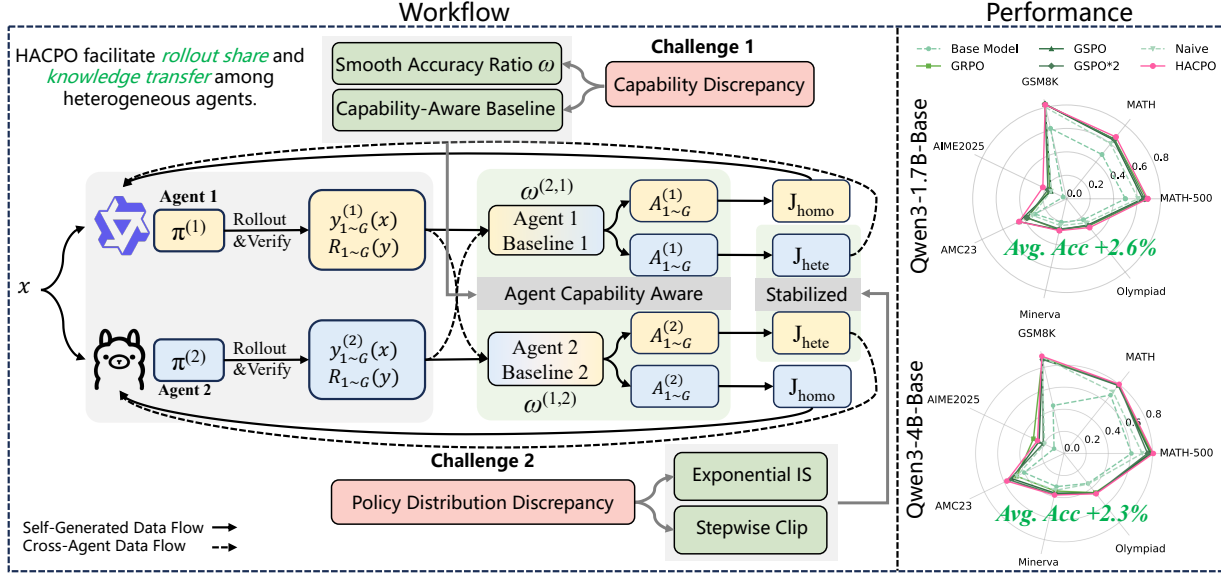


Figure 2. In HACPO, shared rollouts from multiple heterogeneous agents are leveraged for collaborative training. Built upon vanilla RL Optimization, HACPO introduces four algorithmic innovations to mitigate capability and policy distribution discrepancy.

We first formalize this setting as **Heterogeneous Agent Collaborative Reinforcement Learning (HACRL)**, which captures collaborative policy optimization among heterogeneous agents that execute independently at inference time. We then propose **Heterogeneous Agent Collaborative Policy Optimization (HACPO)** to solve HACRL (Figure 2). Compared to vanilla RL optimization, HACPO improves training in two critical aspects: **(1) Maximized Sample Utilization**. In an n -agent system, each rollout can be reused up to n times, substantially improving sample efficiency. **(2) Bidirectional Knowledge Transfer**. By learning from one another, agents acquire complementary knowledge unavailable through self-learning alone, enabling all agents to break performance bottlenecks.

As illustrated in Figure 1, HACRL differs fundamentally from LLM-based multi-agent reinforcement learning (MARL) (Liao et al., 2025b), which trains agents to coordinate and jointly solve tasks through interaction. It is also distinct from knowledge distillation (KD), where a typically fixed teacher transfers knowledge to a student. In contrast, HACRL focuses on *independent execution with collaborative optimization*: agents share verified rollouts during training, yet operate independently at inference time without coordination. In this work, our contributions can be summarized as:

[Problem Definition]. We formulate HACRL as a collaborative policy optimization problem for heterogeneous agents under RLVR, aiming to achieve mutual benefit through cross-agent rollout reuse while controlling distribution shifts caused by heterogeneity.

[Algorithm]. We propose HACPO to address this problem, with four following modifications: **(1) Agent-Capability-**

Aware Advantage Estimation, **(2) Model Capabilities Discrepancy Coefficient**, **(3) Exponential Importance Sampling**, and **(4) Stepwise Clipping**, as shown in Figure 1. These tailored techniques enable the agents to engage in effective and stable mutual learning.

[Performance]. We evaluate HACPO across three types of heterogeneity and seven challenging mathematical reasoning benchmarks, demonstrating consistent performance improvements for all participating agents.

2. Heterogeneous Agent Collaborative Reinforcement Learning

2.1. Heterogeneous LLM Agent Taxonomy

Let π_θ denote a large language model (LLM) agent parameterized by $\theta \in \Theta$, where Θ specifies the complete parameter space, including architecture, dimensionality, and trainable weights. Let V_π denote the output vocabulary of agent π_θ . We consider a collaborative policy optimization setting in which multiple LLM agents are jointly optimized toward a shared or coupled objective.

We categorize heterogeneity among distinct LLM agents into three types: (1) heterogeneous state; (2) heterogeneous size; (3) heterogeneous model.

Definition 2.1 (Heterogeneous State). Two LLM agents $\pi_{\theta_1}^{(1)}$ and $\pi_{\theta_2}^{(2)}$ are said to exhibit *heterogeneous state* if $\Theta_1 = \Theta_2$ and $\dim(\theta_1) = \dim(\theta_2)$, but $\theta_1 \neq \theta_2$ at the start of collaborative policy optimization.

Definition 2.2 (Heterogeneous Size). Two LLM agents $\pi_{\theta_1}^{(1)}$ and $\pi_{\theta_2}^{(2)}$ are said to exhibit *heterogeneous size* if they belong to the same model family and share the same ar-

chitectural design principles, but have different parameter dimensionalities, i.e., $\dim(\theta_1) \neq \dim(\theta_2)$, with $\theta_1 \neq \theta_2$ at the start of collaborative policy optimization.

Definition 2.3 (Heterogeneous Model). Given two LLM agents $\pi_{\theta_1}^{(1)}$ and $\pi_{\theta_2}^{(2)}$, we define them to exhibit *heterogeneous model* heterogeneity if their model architectures differ (e.g., tokenizer, attention mechanism, or training objective), their parameter spaces and sizes are distinct (i.e., $\Theta_1 \neq \Theta_2$), and their initial parameter instantiations are unique (i.e., $\theta_1 \neq \theta_2$).

Remark 2.4. This taxonomy represents increasing degrees of heterogeneity: heterogeneous state differs only in optimization state, heterogeneous size introduces capacity mismatch, and heterogeneous model captures architectural and representational divergence. This hierarchy enables a systematic study of collaborative policy optimization among heterogeneous LLM agents.

2.2. Problem Formalization

We consider the Heterogeneous Agent Collaborative Reinforcement Learning (HACRL) framework with n LLM agents. Each agent $k \in \{1, \dots, n\}$ is associated with a policy π_{θ_k} . All agents operate on a shared task distribution \mathcal{D} and exhibit heterogeneity as defined in Section 2.1.

During training, for a query $q \sim \mathcal{D}$, each agent k independently samples G candidate responses from its policy:

$$Y_k(q) = \{y_{k,1}, \dots, y_{k,G}\} \sim \pi_{\theta_k}(\cdot | q). \quad (1)$$

The joint response set across all agents is $\mathcal{Y}(q) = \bigcup_{k=1}^n Y_k(q)$. Since all agents solve the same task, a shared reward function $R(\cdot)$ is applied to every response. The joint reward set is

$$\mathcal{R}(q) = \{R(y_{k,i}) | k = 1, \dots, n, i = 1, \dots, G\}. \quad (2)$$

For notational convenience, we denote by

$$\mathcal{R}_k(q) = \{R(y_{k,i}) | i = 1, \dots, G\} \quad (3)$$

the rewards corresponding to responses generated by agent k .

Definition 2.5 (HACRL Problem). Consider a system of n heterogeneous agents. For a query $q \sim \mathcal{D}$, let $\mathcal{Y}(q)$ and $\mathcal{R}(q)$ denote the joint response and reward sets, respectively. The objective of *Heterogeneous Agent Collaborative Reinforcement Learning* is to optimize each agent $k \in \{1, \dots, n\}$ by maximizing

$$J^{(k)} = J_{\text{homo}}^{(k)}(Y_k(q), \mathcal{R}_k(q)) + J_{\text{hete}}^{(k)}(\{Y_j(q), \mathcal{R}_j(q)\}_{j \neq k}), \quad (4)$$

where $J_{\text{homo}}^{(k)}$ is computed using rollouts generated by agent k itself, and $J_{\text{hete}}^{(k)}$ leverages rollouts generated by the other agents.

This formulation enables each agent to benefit from both self-generated experiences and cross-agent information under collaborative reinforcement learning.

3. Heterogeneous Agent Collaborative Policy Optimization

In this section, we propose HACPO, a novel multi-agent collaborative optimization framework (Algorithm Procedure is shown in Appendix E): for *one* given task, *multiple* heterogeneous LLM agents execute independently and learn from each other. We summarize the key challenges and corresponding design principles below.

Core Challenge. Facing two challenges shown in Figure 2: the discrepancy of *agent capability* and *policy distribution*, how can an agent effectively learn from rollouts generated by other heterogeneous agents ?

Design Principles. Our method addresses the above challenges through the following components:

- (i) Adaptive Advantage Estimation.** Jointly leveraging self-generated and cross-agent rollouts to perform advantage estimation.
- (ii) Model Capability Discrepancy Awareness.** Assessing the capability discrepancies between agents and reweighting response advantages accordingly.
- (iii) Exponential Importance Sampling.** Computing importance weights to correct for distributional discrepancies across agents.
- (iv) Stepwise Clipping.** Applying stepwise clipping to suppress high-variance or noisy cross-agent responses and stabilize training.

3.1. Agent-Capability-Aware Advantage Estimation

At training step t , for each prompt x , each agent $k \in \{1, \dots, n\}$ generates G responses $\{y_{t,i}^{(k)}\}_{i=1}^G \sim \pi_{\theta_t}^{(k)}(\cdot | x)$. For a single agent, the standard group-relative advantage estimator is

$$A_{t,i}^{(k)}(\text{single}) = \frac{R(y_{t,i}^{(k)}) - \frac{1}{G} \sum_{i=1}^G R(y_{t,i}^{(k)})}{\sigma_t^{(k)}}, \quad (5)$$

where $\sigma_t^{(k)}$ denote the mean and standard deviation of rewards within the group of agent k , respectively.

While Eq. (5) is appropriate for training a single model in isolation, it becomes suboptimal in a multi-agent settings where agents exhibit heterogeneous capabilities. Relying solely on self-generated responses fails to leverage valuable information from other agents, while *naively averaging rewards across all agents disregards inter-model capability*

differences and often results in miscalibrated advantage estimates.

To address this issue, we propose an *agent-capability-aware* advantage estimator. The advantage of response $y_{t,i}^{(k)}$ for agent k is defined as

$$A_{t,i}^{(k)} = \frac{R(y_{t,i}^{(k)}) - \hat{\mu}_t^{(k)}}{\sigma_{t,joint}}, \quad \sigma_{t,joint} = \text{std}\{\mathcal{R}_t(q)\} \quad (6)$$

where $\mathcal{R}_k(q)$ refers to responses from all agents (Eq. ??), the capability-adjusted baseline $\hat{\mu}_t^{(k)}$ is computed by

$$\hat{\mu}_t^{(k)} = \frac{1}{nG} \sum_{j=1}^n \sum_{i=1}^G \omega_t^{(k,j)} R(y_{t,i}^{(j)}). \quad (7)$$

Here, $\omega_t^{(k,j)}$ is a *capability ratio* that rescales responses from agent j when estimating the baseline for agent k , defined as

$$\omega_t^{(k,j)} = \frac{\hat{P}_t^{(k)}}{\hat{P}_t^{(j)}}. \quad (8)$$

The quantity $\hat{P}_t^{(k)}$ denotes a smoothed estimate of the recent performance of agent k , obtained by averaging the per-batch mean rewards over a sliding window of the most recent K steps:

$$\begin{aligned} \hat{P}_t^{(k)} &= \frac{1}{K} \sum_{\tau=t-K+1}^t P_\tau^{(k)} \\ P_\tau^{(k)} &= \frac{1}{G} \sum_{i=1}^G R(y_{\tau,i}^{(k)}). \end{aligned} \quad (9)$$

Intuitively, when estimating the advantage baseline in a group for agent k , rewards from other agents are reweighted according to their relative capabilities, allowing all responses to contribute while preserving agent-specific calibration. The temporal smoothing over the most recent K batches stabilizes the capability estimates and reduces variance. We further show that this advantage estimation is unbiased in Theorem 4.1.

3.2. Model Capabilities Discrepancy Coefficient

To address capability discrepancies across heterogeneous agents, we employ the capability ratio $\omega_t^{(k,j)}$, introduced earlier, as a quantitative measure of relative model competence. When training agent k , advantages computed from samples generated by other agents are rescaled according to their relative capability. This design encourages an agent to learn more aggressively from stronger agents, while adopting a more conservative update when incorporating samples from weaker ones.

Formally, suppose that agent k is updated at training step t using a response $y_{t,i}^{(j)}$ generated by agent j . The effective advantage used for updating agent k is defined as

$$\tilde{A}_{t,i}^{(k)} = \begin{cases} A_{t,i}^{(k)} & y_{t,i}^{(k)} \in \mathcal{D}_t^{(k)} \\ \omega_t^{(j,k)} A_{t,i}^{(j)} & y_{t,i}^{(j)} \in \mathcal{D}_t^{(j)}, j \neq k \end{cases} \quad (10)$$

where $\mathcal{D}_t^{(j)}$ denotes the set of samples generated by agent j at step t . Here, $\omega_t^{(k,j)}$ represents the performance ratio between agents k and j at training step t , with larger values indicating that agent k outperforms agent j .

Remark 3.1. We emphasize that the capability ratio $\omega_t^{(k,j)}$ appears in two distinct but complementary roles in our framework. Together, they enable stable and capability-aware collaboration across heterogeneous agents.

(i) Baseline Calibration. In Section 3.1, $\omega_t^{(k,j)}$ is used to rescale rewards from agent j when estimating the capability-aware baseline $\hat{\mu}_t^{(k)}$. Its role is to *align reward statistics across heterogeneous agents*, ensuring that the baseline used for agent k is properly calibrated.

(ii) Gradient Modulation. In Eq. (10), the same ratio $\omega_t^{(k,j)}$ is applied directly to the advantage of responses generated by agent j when updating agent k . Here, $\omega_t^{(k,j)}$ serves as a *learning-rate-like modulation factor*, amplifying gradients from stronger agents while attenuating those from weaker ones.

3.3. Exponential Importance Sampling

Importance sampling is commonly used to correct distributional mismatches between samples generated by different policies. Following GSPO, we adopt a sequence-level importance ratio and extend it to the heterogeneous multi-agent setting. When updating agent k at step t , for a response $y_{t,i}^{(j)}$ generated by agent j , we define

$$s_{t,i}^{(k,j)} = \left(\frac{\pi_{\theta_t}^{(k)}(y_{t,i}^{(j)})}{\pi_{\theta_{old}}^{(j)}(y_{t,i}^{(j)})} \right)^{\frac{1}{|y_{t,i}^{(j)}|}}. \quad (11)$$

For combinations of heterogeneous agents that satisfy Definition 2.3 with incompatible tokenizers, we detokenize the response into text and retokenize it using the target agent’s tokenizer. Through sequence-level normalization, the slight length discrepancies arising from re-tokenization become negligible.

In heterogeneous settings, inter-agent policy discrepancies can be much larger than on-policy updates, making direct use of this ratio overly aggressive. To mitigate this issue, we introduce a non-gradient exponential reweighting:

$$\tilde{s}_{t,i}^{(k,j)} = s_{t,i}^{(k,j)} \cdot (\text{sg}[s_{t,i}^{(k,j)}])^\alpha \quad k \neq j, s_{t,i}^{(k,j)} < 1.0 \quad (12)$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator and $\alpha \geq 0$ controls the degree of conservativeness.

This design biases agent k toward learning from agents whose output distributions are more aligned with its own, while reducing the impact of large cross-agent distribution shifts.

3.4. Stepwise Clipping

We argue that the cross-agent importance sampling ratio $s_{t,i}^{(k,j)}$ exhibits following fundamentally different behaviors from the self-agent ratio $s_{t,i}^{(k,k)}$:

- (1) $s_{t,i}^{(k,j)}$ evolves dynamically across training iterations;
- (2) Within a single training step, $s_{t,i}^{(k,j)}$ fluctuates irregularly as the number of parameter updates increases, in contrast to the self-agent ratio, which typically decays smoothly.

Additional experimental details on importance sampling in the heterogeneous-agent setting are provided in Appendix C due to space constraints.

Asymmetric Clipping bounds for Cross-Agent. Due to the above distinctions, conventional symmetric clipping of the form $[1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}}]$ is no longer appropriate for cross-agent importance sampling. This is because, unlike self-agent importance sampling, cross-agent importance sampling $s_{t,i}^{(k,j)} > 1$ corresponds to assigning a higher likelihood to responses generated by another agent than to those generated by the current agent itself. Such amplification is undesirable in heterogeneous settings although highly rare, as it may guide cross-agent rollouts to dominate the gradient updates of the current agent, thereby introducing severe distributional bias. Instead, we adopt the following asymmetric clipping scheme, where δ is a hyperparameter that controls the lower clipping bound:

$$s_{t,i}^{(k,j)} \in [1.0 - \delta, 1.0], k \neq j, \quad (13)$$

In Eq. 13, we deliberately limit the upper bound of clipping to 1.0. This simple modification ensures that cross-agent samples can only downweight, but never upweight the learning signals relative to on-policy samples. If $s_{t,i}^{(k,j)} < 1 - \delta$, the corresponding sample is considered too far from the current policy and is discarded. In practice, we typically set $\delta = 0.2$.

Stepwise Clipping. To account for the accumulation of policy drift, we additionally introduce a stepwise clipping strategy within each training step. Let k denote the number of parameter updates performed so far within the current step, and let δ_{step} denote the per-update tightening factor. The clipping operator is defined as

$$\text{clip}(s_{t,i}^{(k,j)}) = \text{clip}\left(s_{t,i}^{(k,j)}, 1 - \delta + k \cdot \delta_{\text{step}}, 1.0\right), \quad (14)$$

where $k \neq j$. Under this scheme, cross-agent responses appearing in later mini-batches are subject to increasingly stricter clipping bounds. This prevents cross-agent rollouts from dominating late-stage updates within a batch, thereby improving training stability in heterogeneous collaborative policy optimization.

4. Theoretical Analysis of HACPO

In this section, we establish the theoretical foundations of HACPO by addressing two fundamental questions: **(i)** Whether the mixed-response advantage baseline introduces systematic bias; **(ii)** Whether learning from cross-agent rollouts yields a valid optimization direction.

4.1. Unbiasedness of Advantage Estimation

We first demonstrate that the proposed *Agent-Capability-Aware Advantage Estimation* in HACPO is unbiased.

Theorem 4.1 (Unbiased Advantage Estimator). *Let agent k generate G responses $\{y_{t,i}^{(k)}\}_{i=1}^G \sim \pi_{\theta_k}(\cdot | q_t)$ at training step t , and let $\mu_t^{(k)}$ denote the mixed-response advantage baseline used by HACPO for agent k . Then, under the shared reward function $R(\cdot)$, the baseline is unbiased in the sense that*

$$\mathbb{E}[\mu_t^{(k)}] = \mathbb{E}_{y \sim \pi_{\theta_k}(\cdot | q_t)}[R(y)],$$

where the expectation is taken over the stochasticity of response generation.

Furthermore, define the advantage for the i -th response of agent k as:

$$A_{t,i}^{(k)} := R(y_{t,i}^{(k)}) - \mu_t^{(k)}. \quad (15)$$

Consequently, the unbiasedness of $A_{t,i}^{(k)}$ is established as follows:

Corollary 4.2 (Unbiased Advantage). *Under the conditions of Theorem 4.1, for any $i \in \{1, \dots, G\}$,*

$$\mathbb{E}[A_{t,i}^{(k)}] = \mathbb{E}_{y \sim \pi_{\theta_k}(\cdot | q_t)}[R(y)] - \mathbb{E}[\mu_t^{(k)}] = 0.$$

Theorem 4.1 states that, although HACPO computes the baseline $\mu_t^{(k)}$ using a mixture of responses collected from multiple agents, this mixed baseline remains *unbiased* for the on-policy expected reward of the trained agent k . This result provides theoretical justification for incorporating cross-agent responses into advantage estimation without introducing systematic bias, as shown in Corollary 4.2.

4.2. Gradient Consistency and Effectiveness

The effectiveness of HACPO relies on the premise that learning from cross-agent rollouts induces an optimization direction consistent with standard on-policy learning. In this section, we formalize this intuition by showing that the gradient of the heterogeneous objective is positively aligned with that of the homogeneous objective.

We analyze the optimization directions induced by the homogeneous objective $\mathcal{J}_{\text{homo}}^{(k)}$ and the heterogeneous objective $\mathcal{J}_{\text{hete}}^{(k)}$ for agent k . Detailed gradient derivations are deferred to Appendix D.2; here, we focus on their directional properties.

For the heterogeneous objective $\mathcal{J}_{\text{hete}}^{(k)}$, HACPO incorporates cross-agent responses through importance weighting, clipping, and capability-aware scaling. Under this design, we establish the following result.

Theorem 4.3 (Gradient Alignment and Effectiveness of HACPO). *For each agent k , the gradients of the heterogeneous and homogeneous objectives are positively aligned, i.e., $\langle \nabla_{\theta_k} \mathcal{J}_{\text{hete}}^{(k)}, \nabla_{\theta_k} \mathcal{J}_{\text{homo}}^{(k)} \rangle > 0$.*

Theorem 4.3 shows that cross-agent responses provide a directionally consistent learning signal. As a result, HACPO preserves the optimization direction of standard on-policy learning while enabling agents to leverage additional cross-agent experience, thereby improving data efficiency without introducing adverse optimization bias. The complete proof is provided in Appendix D.3.

5. Experiment

Setting Details. We adopt 7.5k high quality math questions from the MATH dataset (Hendrycks et al., 2021) for training. During evaluation, we select a comprehensive set of benchmarks: MATH-500, MATH, GSM8K (Cobbe et al., 2021), AIME2025, AMC23 (Cairns, 1916), Minerva (Lewkowycz et al., 2022) and Olympiad (He et al., 2024).

To verify the effectiveness of our method, we conduct experiments on the three heterogeneity settings mentioned in Section 2.1. We compare our approach against the following baselines: (1) **Standard Single-Agent Baselines (GRPO, GSPO)**, which serve as benchmarks for isolated training performance (same rollout cost as HACPO but with half the policy updates); (2) **Resource-Equivalent Baseline (GSPO $\times 2$)**, a single-agent GSPO setting with double rollouts and updates in every step. This serves to rule out the impact of increased data volume and verify the complementary value of heterogeneous agents (double the rollout cost of HACPO but with the same policy updates); (3) **Naive Collaborative Baseline (Naive)**, a two-agent setting with shared rollouts but lacking the algorithmic innovations in

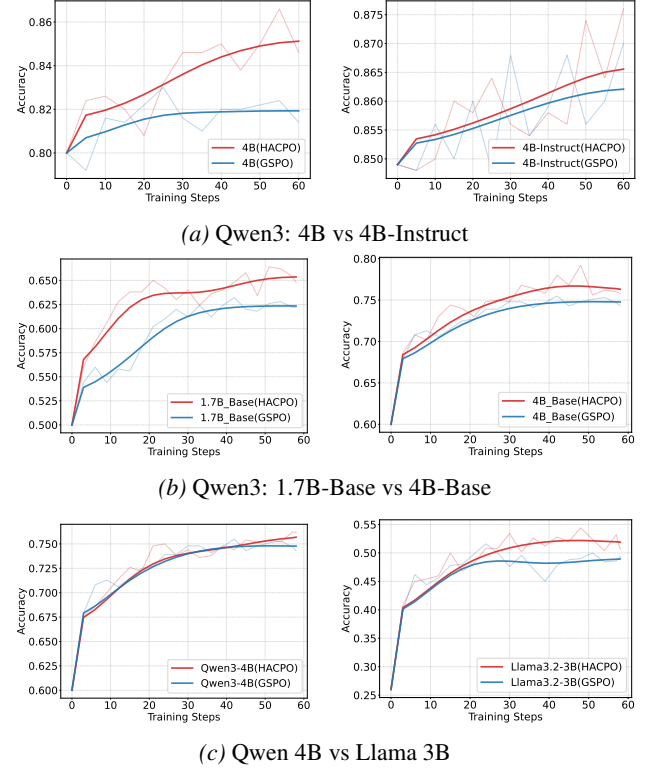


Figure 3. Training curves of GSPO and HACPO

Section 3, used to validate the necessity of our proposed discrepancy mitigation techniques (same rollout and policy update costs as HACPO).

5.1. Result and Analysis

As detailed in Table 1, HACPO demonstrates superior final performance compared to all baselines across various heterogeneous settings. To illustrate the learning dynamics, Figure 3 presents the training curves of HACPO versus the single-agent GSPO baseline. We attribute these performance gains to two primary mechanisms inherent in the HACPO paradigm: (1) Capability-driven guidance, where stronger models assist in enhancing the performance of weaker ones; and (2) Mutual knowledge exchange, which involves the sharing of complementary rollouts—encompassing both correct solutions and informative errors—between agents.

Heterogeneous State. In the Qwen3-4B and Qwen3-4B-Instruct setting, we observe asymmetric but non-trivial gains: while the 4B model improves more substantially, the Instruct model also exhibits consistent performance improvements. Although this setting corresponds to heterogeneous state, where agents differ only due to post-training stages, HACPO still enables the stronger agent to benefit from the weaker one. Specifically, the weaker agent contributes complementary exploration signals—such as alternative reasoning paths and informative errors—that are underrepresented in the stronger agent’s own rollouts. As a result, learning

Table 1. Main results across three heterogeneity settings. We compare our method against Standard Single-Agent Baselines (GRPO, GSPO), a Resource-Equivalent Baseline (GSPO $\times 2$) and a Naive multi-agent rollout share baseline(Naive).

Model	MATH-500	MATH	GSM8K	AIME2025	AMC23	Minerva	Olympiad	AVG
Qwen3-4B + Qwen3-4B-Instruct								
4B	0.802	0.836	0.907	0.335	0.65	0.39	0.524	0.635
4B(GRPO)	0.88	0.889	0.918	0.582	0.775	0.386	0.592	0.717
4B(GSPO)	0.854	0.87	0.925	0.485	0.675	0.412	0.564	0.684
4B(GSPO $\times 2$)	0.876	0.875	0.923	0.522	0.675	0.39	0.579	0.691
4B(Naive)	0.728	0.737	0.891	0.378	0.6	0.353	0.394	0.583
4B(HACPO)	0.91	0.905	0.933	0.622	0.85	0.423	0.643	0.755
4B-Instruct	0.938	0.937	0.936	0.696	0.85	0.441	0.722	0.789
4B-Instruct(GRPO)	0.93	0.933	0.933	0.676	0.875	0.43	0.72	0.785
4B-Instruct(GSPO)	0.938	0.94	0.939	0.72	0.9	0.43	0.726	0.799
4B-Instruct(GSPO $\times 2$)	0.932	0.939	0.942	0.74	0.9	0.43	0.711	0.799
4B-Instruct(Naive)	0.844	0.845	0.936	0.547	0.725	0.39	0.552	0.691
4B-Instruct(HACPO)	0.948	0.943	0.946	0.757	0.95	0.452	0.732	0.813
Qwen3-1.7B-Base + Qwen3-4B-Base								
1.7B-Base	0.5	0.483	0.616	0.033	0.3	0.206	0.229	0.338
1.7B-Base(GRPO)	0.682	0.652	0.824	0.16	0.375	0.272	0.298	0.466
1.7B-Base(GSPO)	0.648	0.641	0.826	0.148	0.45	0.272	0.287	0.467
1.7B-Base(GSPO $\times 2$)	0.664	0.65	0.829	0.177	0.375	0.265	0.293	0.475
1.7B-Base(Naive)	0.608	0.601	0.798	0.147	0.325	0.235	0.263	0.425
1.7B-Base(HACPO)	0.69	0.674	0.822	0.225	0.45	0.279	0.314	0.493
4B-Base	0.61	0.676	0.445	0.1	0.4	0.308	0.347	0.412
4B-Base(GRPO)	0.796	0.788	0.885	0.307	0.475	0.349	0.454	0.579
4B-Base(GSPO)	0.782	0.787	0.877	0.25	0.525	0.368	0.46	0.578
4B-Base(GSPO $\times 2$)	0.756	0.794	0.873	0.208	0.55	0.382	0.463	0.575
4B-Base(Naive)	0.708	0.712	0.895	0.196	0.475	0.342	0.354	0.526
4B-Base(HACPO)	0.808	0.801	0.903	0.267	0.575	0.386	0.467	0.601
Qwen3-4B-Base + Llama3.2-3B-Instruct								
qwen3-4B	0.61	0.676	0.445	0.1	0.4	0.308	0.347	0.412
qwen3-4B(GRPO)	0.796	0.788	0.885	0.307	0.475	0.349	0.454	0.579
qwen3-4B(GSPO)	0.782	0.787	0.877	0.25	0.525	0.368	0.46	0.578
qwen3-4B(GSPO $\times 2$)	0.756	0.794	0.873	0.208	0.55	0.382	0.463	0.575
qwen3-4B(Naive)	0.734	0.712	0.895	0.143	0.55	0.342	0.354	0.526
qwen3-4B(HACPO)	0.786	0.783	0.921	0.268	0.6	0.379	0.442	0.597
llama3.2-3B	0.267	0.441	0.788	0.0	0.2	0.169	0.158	0.289
llama3.2-3B(GRPO)	0.502	0.507	0.814	0.0	0.25	0.199	0.174	0.349
llama3.2-3B (GSPO)	0.512	0.501	0.812	0.054	0.225	0.184	0.17	0.351
llama3.2-3B (GSPO $\times 2$)	0.488	0.498	0.829	0.0	0.175	0.188	0.159	0.334
llama3.2-3B (Naive)	0.406	0.407	0.734	0.0	0.225	0.177	0.107	0.294
llama3.2-3B (HACPO)	0.566	0.548	0.826	0.054	0.35	0.176	0.208	0.39

is not purely unidirectional. Even when capability-driven guidance dominates, the stronger agent can still extract useful supervisory signals from the weaker agent, leading to measurable performance gains.

Heterogeneous Size. In the Qwen3-1.7B-Base and Qwen3-4B-Base setting, both models improve significantly, validating the mechanism of mutual knowledge exchange. Even with lower capability, the 1.7B model serves as a distinct explorer, generating valuable erroneous responses and a few unique correct solutions that the 4B model fails to produce, thereby facilitating bidirectional knowledge transfer.

Heterogeneous Model. Finally, we consider the heterogeneous model setting involving Qwen3-4B-Base and Llama3.2-3B-Instruct, which differ substantially in archi-

itecture, tokenizer, and training objectives. Despite this high degree of heterogeneity, we observe consistent performance improvements in both models. These results demonstrate that HACPO is able to extract transferable knowledge from cross-model rollouts and effectively share it across heterogeneous agents. By leveraging verified responses—including correct solutions and informative failure cases—each model can learn from complementary reasoning patterns that are absent from its own policy distribution.

The experimental results show that HACPO significantly improves performance across all three types of heterogeneity, validating its generality and robustness. Additionally, the differences observed among the three settings shed light on the two underlying mechanisms of HACPO.

Table 2. Ablation of Advantage Estimator

Model	MATH-500	math	gsm8k	aime2025	ACM23	minerva	olympiad	AVG
1.7B(HACPO - Adv)	0.696	0.659	0.825	0.126	0.375	0.261	0.313	0.465
1.7B(HACPO)	0.69	0.674	0.822	0.225	0.45	0.279	0.314	0.493
4B(HACPO - Adv)	0.774	0.771	0.912	0.308	0.55	0.348	0.442	0.586
4B(HACPO)	0.808	0.801	0.903	0.267	0.575	0.386	0.467	0.601

Table 3. Ablation of Model Capabilities Discrepancy Coefficient

Model	MATH-500	math	gsm8k	aime2025	ACM23	minerva	olympiad	AVG
1.7B(HACPO - ω)	0.666	0.657	0.806	0.105	0.425	0.25	0.324	0.462
1.7B(HACPO)	0.69	0.674	0.822	0.225	0.45	0.279	0.314	0.493
4B(HACPO - ω)	0.816	0.797	0.902	0.261	0.55	0.401	0.475	0.6
4B(HACPO)	0.808	0.801	0.903	0.267	0.575	0.386	0.467	0.601

5.2. Ablation Study

Agent-Capability-Aware Advantage Estimation. Ablation on the Qwen3-1.7B/4B-Base combination (Table 2) confirms that removing this module significantly degrades performance. This decline stems from the systematic bias in standard group-relative advantages in multi-agent setting due to the capability discrepancy cross heterogeneous agents. Our method addresses this by constructing *agent-capability-aware* advantage baselines—raising the standard for the stronger models and lowering it for the weaker ones—thereby preserving the unbiasedness of the advantage estimator established in Theorem 4.1.

Model Capabilities Discrepancy Coefficient. We isolate the role of this coefficient in gradient modulation by disabling it in the policy update (Eq. 10) while retaining it for advantage estimation. Table 3 confirms that removing this modulation degrades performance. This validates the coefficient’s critical function as a capability-aware scaler: it amplifies gradients from stronger agents to accelerate learning, while attenuating updates from weaker ones to mitigate potential noise.

Exponential Importance Sampling. We examined the impact of α on Qwen3-1.7B/4B-Base and Qwen3-4B/8B-Base combinations (Table 4). Results highlight a critical trade-off: increasing α enforces a more conservative policy towards cross-agent responses, which aids stability by suppressing large distribution shifts but hinders efficiency by reducing the effective learning signal. Thus, the optimal α is model combination dependent, necessitating a balance between stable convergence and maximal information extraction.

Stepwise Clipping. We assess the necessity of this mechanism on the Qwen3-4B/8B-Base combination. As visualized in Figure 4, removing the clipping constraint (**no Clip**) causes severe instability, while omitting the stepwise schedule (**no Stepwise**) leads to suboptimal convergence compared to the full HACPO. This confirms that the stepwise clipping is indispensable for stabilizing collaborative learning, as neither unconstrained nor statically bounded up-

Table 4. Qwen3-1.7B-Base and Qwen3-4B-Base

α	0.0	1.0	2.0	3.0
Qwen3-1.7B-Base and Qwen3-4B-Base				
1.7B-Base	0.63	0.664	0.654	0.668
4B-Base	0.756	0.792	0.768	0.77
Qwen3-4B-Base and Qwen3-8B-Base				
4B-Base	0.772	0.776	0.77	0.776
8B-Base	0.764	0.772	0.766	0.778

dates suffice to handle high-variance cross-agent responses.

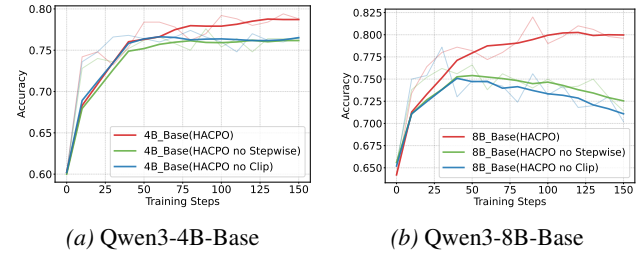


Figure 4. The Ablation of Stepwise Clipping

6. Related Work

Our work is most closely related to Reinforcement Learning with Verifiable Rewards (RLVR), with Group Sequence Policy Optimization (GSPO) being the most relevant prior study. GSPO demonstrates the efficacy of sequence-level importance sampling in Mixture-of-Experts (MoE) models, where tokens may originate from different networks. This insight inspires our approach to facilitate rollout sharing among heterogeneous agents. Additionally, our work shares conceptual parallels with Multi-Agent Reinforcement Learning (MARL). A more detailed discussion of related work is provided in Appendix B.

7. Conclusion

We propose HACRL, a collaborative multi-agent reinforcement learning paradigm tailored for heterogeneous agent

ecosystems. HACRL enables principled rollout sharing among heterogeneous agents, improving sample utilization efficiency while promoting cross-agent knowledge transfer. To instantiate this paradigm, we introduce HACPO, which incorporates four tailored mechanisms to mitigate capability discrepancies and policy distribution shifts arising during collaborative policy optimization. We provide the theoretical analysis establishing the unbiasedness of the proposed advantage estimation scheme and the validity of the resulting optimization direction under controlled heterogeneity. Extensive experiments demonstrate that HACPO consistently and significantly improves performance across all heterogeneity types.

Impact Statement

This paper presents a collaborative policy optimization framework for heterogeneous Large Language Models, aiming to enhance the efficiency and effectiveness of post-training through cross-agent rollout sharing and verifiable rewards. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

- Agarwal, R., Vieillard, N., Zhou, Y., Stanczyk, P., Garea, S. R., Geist, M., and Bachem, O. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024a.
- Agarwal, R., Vieillard, N., Zhou, Y., Stanczyk, P., Garea, S. R., Geist, M., and Bachem, O. C. In *The Twelfth International Conference on Learning Representations*, 2024b.
- Anil, R., Perea, G., Passos, A., Ormandi, R., Dahl, G. E., and Hinton, G. E. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018.
- Cai, W., Jiang, J., Wang, F., Tang, J., Kim, S., and Huang, J. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*, 2025.
- Cairns, W. The mathematical association of america. *The American Mathematical Monthly*, 23(1):1–6, 1916.
- Chen, Z., Ai, T., Li, Y., Li, G., Wei, Y., Zhou, W., Li, G., Yu, B., Chen, Z., Sun, H., Zhuang, F., Li, J., Wang, D., and Ban, Y. Llmboost: Make large language models stronger with boosting, 2025. URL <https://arxiv.org/abs/2512.22309>.
- Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*, 2023.
- Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- Gou, J., Yu, B., Maybank, S. J., and Tao, D. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819, 2021.
- Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- He, C., Luo, R., Bai, Y., Hu, S., Thai, Z., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.
- Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.
- Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- Ho, N., Schmid, L., and Yun, S.-Y. Large language models are reasoning teachers. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 14852–14882, 2023.
- Hsieh, C.-Y., Li, C.-L., Yeh, C.-K., Nakhost, H., Fujii, Y., Ratner, A., Krishna, R., Lee, C.-Y., and Pfister, T. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, 2023.
- Huang, Z., Ban, Y., Fu, L., Li, X., Dai, Z., Li, J., and deqing wang. Adaptive batch-wise sample scheduling for direct preference optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=8FN25Plkts>.

- Kuba, J. G., Chen, R., Wen, M., Wen, Y., Sun, F., Wang, J., and Yang, Y. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251*, 2021.
- Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.
- Li, Y., Zhang, Y., and Sun, L. Metaagents: Simulating interactions of human behaviors for llm-based task-oriented coordination via collaborative generative agents. *arXiv preprint arXiv:2310.06500*, 2023.
- Liao, J., Wen, M., Wang, J., and Zhang, W. Marft: Multi-agent reinforcement fine-tuning. *arXiv preprint arXiv:2504.16129*, 2025a.
- Liao, J., Wen, M., Wang, J., and Zhang, W. Marft: Multi-agent reinforcement fine-tuning. *arXiv preprint arXiv:2504.16129*, 2025b.
- Liu, S., Liang, Z., Lyu, X., and Amato, C. Llm collaboration with multi-agent reinforcement learning. *arXiv preprint arXiv:2508.04652*, 2025.
- Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.
- Ma, H., Hu, T., Pu, Z., Boyin, L., Ai, X., Liang, Y., and Chen, M. Coevolving with the other you: Fine-tuning llm with sequential cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 37:15497–15525, 2024.
- Madaan, L., Didolkar, A., Gururangan, S., Quan, J., Silva, R., Salakhutdinov, R., Zaheer, M., Arora, S., and Goyal, A. Rethinking thinking tokens: Llms as improvement operators. *arXiv preprint arXiv:2510.01123*, 2025.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022a.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022b.
- Park, C., Han, S., Guo, X., Ozdaglar, A., Zhang, K., and Kim, J.-K. Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning. *arXiv preprint arXiv:2502.18439*, 2025.
- Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36: 53728–53741, 2023.
- Rashid, T., Samvelyan, M., De Witt, C. S., Farquhar, G., Foerster, J., and Whiteson, S. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.
- Romero, A. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.
- Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. Hybridflow: A flexible and efficient rlhf framework. *CoRR*, abs/2409.19256, 2024. doi: 10.48550/ARXIV.2409.19256. URL <https://arxiv.org/abs/2409.19256>.
- Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.
- Wan, Z., Li, Y., Wen, X., Song, Y., Wang, H., Yang, L., Schmidt, M., Wang, J., Zhang, W., Hu, S., et al. Rema: Learning to meta-think for llms with multi-agent reinforcement learning. *arXiv preprint arXiv:2503.09501*, 2025.
- Wang, J., Liu, R., Lin, L., Hu, W., Li, X., Zhang, F., Zhou, G., and Gai, K. Aspo: Asymmetric importance sampling policy optimization. *arXiv preprint arXiv:2510.06062*, 2025.
- Weyssow, M., Zhou, X., Kim, K., Lo, D., and Sahraoui, H. Exploring parameter-efficient fine-tuning techniques

- for code generation with large language models. *ACM Transactions on Software Engineering and Methodology*, 34(7):1–25, 2025.
- Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.
- Yang, F., Chen, Z., Wang, X., Lu, X., Chai, J., Yin, G., Lin, W., Ma, S., Zhuang, F., Wang, D., Yang, Y., Li, J., and Ban, Y. Your group-relative advantage is biased, 2026a. URL <https://arxiv.org/abs/2601.08521>.
- Yang, F., Chen, Z., Wang, X., Lu, X., Chai, J., Yin, G., Lin, W., Ma, S., Zhuang, F., Wang, D., et al. Your group-relative advantage is biased. *arXiv preprint arXiv:2601.08521*, 2026b.
- Yang, S., Dou, C., Guo, P., Lu, K., Ju, Q., Deng, F., and Xin, R. Dcpo: Dynamic clipping policy optimization. *arXiv preprint arXiv:2509.02333*, 2025b.
- Yu, C., Velu, A., Vinitisky, E., Gao, J., Wang, Y., Bayen, A., and Wu, Y. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.
- Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
- Zhao, Y., Liu, Y., Liu, J., Chen, J., Wu, X., Hao, Y., Lv, T., Huang, S., Cui, L., Ye, Q., et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025.
- Zheng, C., Liu, S., Li, M., Chen, X.-H., Yu, B., Gao, C., Dang, K., Liu, Y., Men, R., Yang, A., et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.
- Zhong, Y., Kuba, J. G., Feng, X., Hu, S., Ji, J., and Yang, Y. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32):1–67, 2024.
- Zou, J., Ban, Y., Li, Z., Qi, Y., Qiu, R., Yang, L., and He, J. Transformer copilot: Learning from the mistake log in LLM fine-tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=MRvxlTlkNQ>.

A. Training and Evaluation Details

All experiments in this paper are conducted using verl (Sheng et al., 2024). In the experiments, we set the maximum prompt length to 1024 and the maximum response length to 4096. We use the MATH dataset for training. The learning rate is set to 1×10^{-6} . For the responses generated by the trained agents in HACPO or single GSPO, we set $\epsilon_{\text{low}} = 0.0003$ and $\epsilon_{\text{high}} = 0.0004$, which is consistent with the setting mentioned in GSPO (Zheng et al., 2025). As for the single GRPO, we set $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$, which follows the trick mentioned in DAPO (Yu et al., 2025) and is widely used. The batch size is set to 128, with a mini-batch size of 64 and $n = 8$ rollouts per prompt. The total batch size is set to 128, with a mini-batch size of 64 and $n = 8$ rollouts per prompt. In the Resource-Equivalent Baseline (GSPO $\times 2$), we use a mini-batch size of 32 and $n = 16$ rollouts per prompt to ensure double updates per step, while maintaining a consistent number of rollouts per update with other settings. We train for one epoch, except when examining the impact of stepwise clipping on stabilizing the training process. During evaluation, due to the high complexity of benchmarks such as AIME2025, we adopt a maximum response length of 8196 tokens in the main experiments and the ablation of Agent-Capability-Aware Advantage Estimator and Model Capability Discrepancy Coefficient (Table 1, 7, 3 2). For all other ablation studies, the maximum response length is kept consistent with the training configuration and is set to 4096 tokens. For the main experimental results, we report best@30 on AIME2025, while avg@1 is used for all other benchmarks. Our experiment is conducted on eight GPUs.

Regarding the models used in our experiments, We employed the Qwen3 (Yang et al., 2025a) and Llama3.2 (Grattafiori et al., 2024) series of models. In detail, Qwen3-(1.7B/4B/8B)-Base denotes the base models, while Qwen3-(1.7B/4B/8B) refer to the distilled variants obtained through strong model distillation from their corresponding base models. In addition, Qwen3-4B-Instruct is a further fine-tuned version of Qwen3-4B, designed to better follow user instructions and generate more accurate responses.

In the parameter design of HACPO, when evaluating model capabilities, we use the results from the most recent K batches to perform smoothing. In all experiments, we set $K = 5$. For the clipping boundary δ in the exponential importance sampling of α , as well as the gradient clipping step size δ_{step} , each experiment has slight variations. We provide the specific settings used for each experiment in the Table 5. A commonly used set of parameters is $\alpha = 1$, $\delta = 0.8$, and $\delta_{\text{step}} = 0.025$.

Table 5. The Details of Hyperparameter

Model Combination	α	δ	δ_{step}
qwen3-4B + qwen3-4B-Instruct	3.0	0.8	0.01
qwen3-1.7B-Base + qwen3-4B-Base	1.0	0.8	0.025
qwen3-4B-Base + qwen3-8B-Base	3.0	0.8	0.025
llama3.2-1B-Instruct + llama3.2-3B-Instruct	1.0	0.9	0.01
qwen3-1.7B-Base + llama3.2-1B-Instruct	1.0	0.8	0.025
qwen3-4B-Base + llama3.2-3B-Instruct	1.0	0.8	0.025

B. Additional Related Work

B.1. Reinforcement Learning From Verifiable Rewards

GRPO is one of the main algorithms used in Reinforcement Learning From Verifiable Rewards, and (Yang et al., 2026a) provides a principled theoretical analysis of group-based advantage estimation. The primary modification of GRPO (Shao et al., 2024) involves the formation of a set of responses generated from the same prompt, within which the advantage for each response is computed. This approach eliminates the need for a critic network, thereby significantly reducing both memory and computational overhead. Several variants of GRPO (Yu et al., 2025; Yang et al., 2025b; Zhao et al., 2025; Wang et al., 2025) have been proposed to address issues in GRPO, the most related one is GSPO (Zheng et al., 2025), which improve the performance and generalization of GRPO.

GSPO replaces the token-level importance sampling ratio in GRPO with a sequence-level ratio. GSPO demonstrates greater suitability than GRPO for fine-tuning Mixture-of-Experts (MoE) models. During inference, MoE models dynamically activate different expert networks (Cai et al., 2025). When employing GRPO, if the current policy and the sampling policy activate different experts for a given token, the importance sampling weight for that token can become an outlier, leading to training instability. In contrast, GSPO averages the importance sampling ratio across all tokens within the response, thereby significantly enhancing stability. Importance sampling essentially acts as a weighting mechanism to diminish the gradient contributions from samples that deviate substantially from the current policy’s distribution. The sequence-level importance

sampling employed by GSPO proves particularly effective for MoE models with varying expert networks. This success inspires a broader consideration of measuring the deviation between a sample from other models and the current policy distribution.

Traditional RLVR methods like GRPO and GSPO optimize agents independently, often leading to costly on-policy sampling and underutilized intermediate rollouts. **HACPO** builds upon these group-based paradigms by enabling cross-agent rollout sharing. It maximizes sample utilization by allowing each rollout in an n -agent system to be leveraged up to n times, directly addressing the efficiency bottlenecks of isolated RLVR training.

B.2. Multi-Agent Reinforcement Learning (MARL)

Multi-Agent Reinforcement Learning (MARL) represents a paradigm in Reinforcement Learning (RL), where multiple agents evolve collectively (Lowe et al., 2017; Kuba et al., 2021; Yu et al., 2022; Zhong et al., 2024; Rashid et al., 2020; Foerster et al., 2018). MARL has gradually been applied to LLM-based agent scenarios. Most works in MARL focus on employing multiple agents to build a comprehensive system, where the agents collaborate to accomplish tasks (Liao et al., 2025a; Park et al., 2025; Wan et al., 2025; Liu et al., 2025; Li et al., 2023; Du et al., 2023). These works primarily focus on constructing a holistic system in which agents collaborate to accomplish tasks. In contrast, our work targets scenarios in which multiple agents are required to perform tasks independently. Although these works address different settings compared to ours, they still provide valuable inspiration: even when using only the output text as an input prompt, different models can learn from each other. The model’s sampling not only includes the generated text but also the corresponding probability distribution information. By directly utilizing these samples for policy updates, rather than as inputs, the model can more effectively learn the knowledge of other models.

Several works have used MARL frameworks to fine-tune models. For example, in COPY (Ma et al., 2024), two copies of the same model are assigned as the pioneer and the observer, respectively, with the input of the pioneer serving as the output of the observer. The roles are then exchanged to further facilitate knowledge transfer. However, homogeneous models struggle to transcend their intrinsic performance ceilings (Madaan et al., 2025). Besides, such fine-tuning approaches require numerous sampling iterations, leading to low utilization efficiency. Furthermore, using the same model makes it difficult to inject knowledge beyond the model’s intrinsic capabilities.

While MARL typically focuses on collaborative execution where multiple agents coordinate to solve a task jointly, **HACPO** introduces a distinct paradigm: independent execution with collaborative optimization. By facilitating mutual knowledge transfer during training while ensuring agents act independently at inference, HACPO bridges the gap between collective learning benefits and the practical need for autonomous agent operation.

B.3. Knowledge Distillation (KD)

Knowledge Distillation (KD) is a widely adopted technique in the field of Large Language Models (LLMs), where a high-capacity teacher model is utilized to guide the training of a more compact student model (Hinton et al., 2015; Gou et al., 2021; Sanh et al., 2019). The core mechanism involves the teacher conveying not just its final predictions but its nuanced output distribution (dark knowledge), enabling the student to mimic the teacher’s internal logic and probabilistic insights (Hinton et al., 2015; Romero, 2014).

Beyond traditional static methods, recent advancements have transitioned the distillation process from offline to online and on-policy settings (Anil et al., 2018; Agarwal et al., 2024a; Gou et al., 2021; Agarwal et al., 2024b). These approaches allow for the dynamic transfer of knowledge, often leveraging the student’s own generated trajectories to bridge the distribution gap between models. In the context of LLMs, distillation has also evolved into Black-box Distillation, where students learn from the teacher’s generated responses or chain-of-thought rationales when model weights are inaccessible (Hsieh et al., 2023; Ho et al., 2023). The distinction between distillation and our approach lies in the fact that, in our method, there are no “teacher” or “student” models; instead, all models can learn from each other simultaneously. Furthermore, our approach enables models to engage in both self-exploration and learning from other models concurrently.

Standard Knowledge Distillation (KD) relies on a fixed, one-way path where a student mimics a stronger teacher, potentially limiting the system’s ceiling. **HACPO** transcends this by treating heterogeneous agents as peer co-learners. Through Agent-Capability-Aware Advantage Estimation and bidirectional transfer, it allows even weaker models to contribute unique exploration trajectories, facilitating a mutual performance boost that self-learning or one-way distillation cannot achieve.

C. Heterogeneous Agent Importance Sampling Analysis

In the reinforcement learning paradigm, importance sampling is commonly used to stabilize updates, often through a clipping mechanism. The clipping range typically centers around 1.0. For instance, in GSPO, the upper and lower bounds for clipping are set to 1.0004 and 0.9997, respectively. However, in a multi-agent setting, the importance sampling values for samples from other agents do not exhibit the same pattern and fluctuate as training progresses.

In the experiment involving Qwen3-1.7B-Base and Qwen3-4B-Base, we distinguish between self-generated responses and cross-agent responses, denoted as s^{homo} and s^{hete} , respectively. These values represent the average importance sampling across each training step. It is important to note that while s^{homo} remains stable and tends to stay around 1 throughout training, s^{hete} does not follow a fixed range and fluctuates as training progresses. The results are shown in Table 6

Table 6. s^{homo} and s^{hete} of Qwen3-1.7B-Base in all steps

Model	mean	max	min	range
s^{homo}	1.00002	1.00020	0.99960	0.00060
s^{hete}	0.89550	0.93615	0.86198	0.07417

For self-generated responses, as the number of updates(mini batches) within a batch increases, the discrepancy between the sampling policy $\pi_{\text{old}}(\theta)$ and the current policy $\pi(\theta)$ grows, leading to an increased s^{homo} and a higher ratio of clipped tokens. However, for cross-agent responses, the discrepancy between the current policy $\pi^{(k)}(\theta)$ and the sampling model’s policy $\pi_{\text{old}}^{(j)}(\theta)$ fluctuates unpredictably, leading to a variable s^{hete} and the ratio of clipped tokens.

In a batch with multiple mini-batches, as the number of updates increases, self-generated responses become more heavily clipped in later mini-batches due to the growing discrepancy between the current and old policies. Therefore, the influence of cross-agent responses is likely to increase in later mini-batches, as their importance sampling values are less predictable, leading to an instability if they dominate the update.

D. Theoretical Analysis

D.1. Proof of the Unbiasedness of the Advantage Estimator

In this section, we formally establish that the Agent-Capability-Aware advantage estimator introduced in HACPO provides an unbiased estimation of the baseline, equivalent in expectation to the standard single-agent baseline.

Assumption D.1 (Ideal Capability Ratio). While the practical algorithm estimates $\omega_t^{(k,j)}$ using a moving average that includes the current batch to strictly track non-stationary policy changes, we assume that the capability ratio $\omega^{(k,j)}$ is an estimator of the true performance ratio that is statistically independent of the specific stochastic realization of rewards in the current batch.

$$\mathbb{E}_{\{y_{t,i}^{(j)}\}_{i=1}^G \sim \pi_{\theta_j}} \left[\omega^{(k,j)} \cdot R(y_{t,i}^{(j)}) \right] = \mathbb{E}_{\{y_{t,i}^{(k)}\}_{i=1}^G \sim \pi_{\theta_k}} \left[R(y_{t,i}^{(k)}) \right] \quad (16)$$

Remark D.2. Justification: As the sliding window size K increases, the contribution of the current batch to ω diminishes ($\mathcal{O}(1/K)$), thereby asymptotically satisfying the independence assumption. However, excessively large K introduces estimation bias due to the non-stationarity of evolving policies. In practice, we select a finite K as a necessary trade-off: sufficiently large to approximate independence, yet responsive enough to track dynamic capability changes without significant lag.

Theorem D.3 (Unbiasedness of Coupled Baseline). *Consider a set of heterogeneous agents. The expected value of the capability-aware baseline $\hat{\mu}^{(k)}$ computed using samples from multiple agents is equivalent to the expected reward of agent k*

computed solely from its own samples.

$$\mathbb{E}[\mu_t^{(k)}] = \mathbb{E}_{\{y_{t,i}^{(k)}\}_{i=1}^G \sim \pi_{\theta_k}}[R(y_{t,i}^{(k)})] \quad (17)$$

Proof. Without loss of generality, consider the case of two agents, $k = 1$ and $j = 2$. The capability-aware baseline for agent 1 is given by:

$$\mu_t^{(1)} = \frac{1}{2G} \sum_{i=1}^G R(y_{t,i}^{(1)}) + \frac{\omega^{(1,2)}}{2G} \sum_{i=1}^G R(y_{t,i}^{(2)}). \quad (18)$$

Taking the expectation with respect to the policies π_{θ_1} and π_{θ_2} :

$$\mathbb{E}[\mu_t^{(1)}] = \mathbb{E}_{\{y_{t,i}^{(1)}\} \sim \pi_{\theta_1}, \{y_{t,i}^{(2)}\} \sim \pi_{\theta_2}} \left[\frac{1}{2G} \sum_{i=1}^G R(y_{t,i}^{(1)}) + \frac{\omega^{(1,2)}}{2G} \sum_{i=1}^G R(y_{t,i}^{(2)}) \right] \quad (19)$$

$$= \frac{1}{2} \mathbb{E}_{y_{t,i}^{(1)} \sim \pi_{\theta_1}}[R(y_{t,i}^{(1)})] + \frac{\omega^{(1,2)}}{2} \mathbb{E}_{y_{t,i}^{(2)} \sim \pi_{\theta_2}}[R(y_{t,i}^{(2)})]. \quad (20)$$

Invoking Assumption D.1, we treat $\omega^{(1,2)}$ as independent of the current batch's reward realization $R(y^{(2)})$. This allows us to factorize the expectation:

$$\begin{aligned} \mathbb{E}[\mu_t^{(1)}] &= \frac{1}{2} \mathbb{E}_{y_{t,i}^{(1)} \sim \pi_{\theta_1}}[R(y_{t,i}^{(1)})] + \frac{1}{2} \mathbb{E}_{y_{t,i}^{(1)} \sim \pi_{\theta_1}}[R(y_{t,i}^{(1)})] \\ &= \mathbb{E}_{y_{t,i}^{(1)} \sim \pi_{\theta_1}}[R(y_{t,i}^{(1)})]. \end{aligned} \quad (21)$$

Thus, we can obtain the Theorem D.3. \square

Proof of Corollary 4.2.

Proof. By linearity of expectation and the definition in (15),

$$\mathbb{E}[A_{t,i}^{(k)}] = \mathbb{E}[R(y_{t,i}^{(k)})] - \mathbb{E}[\mu_t^{(k)}].$$

Since $y_{t,i}^{(k)} \sim \pi_{\theta_k}(\cdot | q_t)$, we have $\mathbb{E}[R(y_{t,i}^{(k)})] = \mathbb{E}_{y \sim \pi_{\theta_k}(\cdot | q_t)}[R(y)]$. Applying Theorem 4.1 yields $\mathbb{E}[A_{t,i}^{(k)}] = 0$, which proves the claim. \square

D.2. Gradient Analysis

For notational convenience, define the reference direction

$$\mathbf{v} := \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_{\theta_k}} \left[\frac{1}{|y|} \hat{A}^{(k)}(x, y) \nabla_{\theta_k} \log \pi_{\theta_k}(y | x) \right], \quad (22)$$

where $\hat{A}^{(k)}(x, y)$ denotes the advantage signal used to update agent k .

The homogeneous objective $\mathcal{J}_{\text{homo}}^{(k)}$ coincides with the GSPO objective (Zheng et al., 2025). As a consequence, its gradient satisfies:

$$\nabla_{\theta_k} \mathcal{J}_{\text{homo}}^{(k)} \approx \mathbf{v} \quad (23)$$

For the heterogeneous objective $\mathcal{J}_{\text{hete}}^{(k)}$, HACPO incorporates cross-agent responses through importance weighting, clipping, and capability-aware scaling. Using the Importance Sampling Lemma (Lemma D.10) and the non-negativity of the effective reweighting terms, we show that the heterogeneous gradient admits the same reference direction:

$$\langle \nabla_{\theta_k} \mathcal{J}_{\text{hete}}^{(k)}, \mathbf{v} \rangle > 0. \quad (24)$$

We analyze the gradients of the HACPO objective, decomposing it into self-generated ($\mathcal{J}_{\text{homo}}$) and cross-agent ($\mathcal{J}_{\text{hete}}$) components.

Definition D.4 (Sequence-Level Importance Sampling). The sequence level importance sampling ratio is defined as:

$$s_i(\theta) = \left(\frac{\prod_{t=1}^{|y|} \pi_\theta(y_{i,t}|x, y_{<t})}{\prod_{t=1}^{|y|} \pi_{\theta_{old}}(y_{i,t}|x, y_{<t})} \right)^{\frac{1}{|y|}}. \quad (25)$$

D.2.1. HOMOGENEOUS GRADIENT

For the homogeneous component \mathcal{J}_{homo} , the gradient derivation follows the standard GSPO formulation.

Proposition D.5. *The gradient of the homogeneous objective is given by:*

$$\nabla_\theta \mathcal{J}_{homo} = \mathbb{E}_{y_i \sim \pi_{\theta_{old}}^{(1)}} \left[\frac{1}{G} \sum_{i=1}^G s_i(\theta) \hat{A}_i \cdot \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \nabla_\theta \log \pi_\theta(y_{i,t}|x, y_{<t}) \right]. \quad (26)$$

Proof. Starting from the objective $\mathcal{J}_{GSPO} = \mathbb{E}_{y \sim \pi_{old}} [s_i(\theta) \hat{A}_i]$, we apply the log-derivative trick:

$$\nabla_\theta s(\theta) = \nabla_\theta \exp \left(\frac{1}{|y|} \left(\log \left(\prod_{t=1}^{|y|} \pi_\theta(y_{i,t}|x, y_{<t}) \right) - \log \left(\prod_{t=1}^{|y|} \pi_{\theta_{old}}(y_{i,t}|x, y_{<t}) \right) \right) \right) \quad (27)$$

$$= s(\theta) \cdot \frac{1}{|y|} \sum_{t=1}^{|y|} \nabla_\theta \log \pi_\theta(y_{i,t}|x, y_{<t}). \quad (28)$$

Substituting this into the gradient of the expectation yields the result. \square

D.2.2. HETEROGENEOUS GRADIENT

For the heterogeneous component, we consider agent 1 learning from agent 2. The objective utilizes the exponential importance sampling weight.

Proposition D.6. *The gradient of the heterogeneous objective \mathcal{J}_{hete} with respect to θ_1 is:*

$$\nabla_{\theta_1} \mathcal{J}_{hete} = \mathbb{E}_{y_i \sim \pi_{\theta_{old}}^{(2)}} \left[\omega^{(2,1)} \cdot \text{sg}(s_i^{hete}(\theta_1, \theta_2))^{\alpha+1} \hat{A}_i \cdot \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \nabla_{\theta_1} \log \pi_{\theta_1}(y_{i,t}|x, y_{<t}) \right]. \quad (29)$$

Proof. Let $s_i^{hete}(\theta_1, \theta_2) = \left(\frac{\pi_{\theta_1}^{(1)}(y_i)}{\pi_{\theta_2}^{(2)}(y_i)} \right)^{\frac{1}{|y_i|}}$. The objective is defined as:

$$\mathcal{J}_{hete} = \mathbb{E}_{y \sim \pi_{old}} \left[\omega^{(2,1)} \cdot \text{sg}(s^{hete})^\alpha \cdot s^{hete} \cdot \hat{A} \right]. \quad (30)$$

Noting that $\text{sg}[\cdot]$ denotes the stop-gradient operator and $\pi_{old}^{(2)}$ is independent of θ_1 , the gradient acts only on the term $s_i^{hete}(\theta_1, \theta_2)$. Using the derivative property derived in Proof D.2.1, $\nabla_{\theta_1} s_i^{hete}(\theta_1, \theta_2) = s_i^{hete}(\theta_1, \theta_2) \frac{1}{|y_i|} \nabla_{\theta_1} \log \pi_{\theta_1}^{(1)}(y_i)$. Substituting this yields the proposition. \square

D.3. Proof of the Effectiveness of HACPO

In this section, we formally establish the effectiveness of HACPO by demonstrating that the heterogeneous objective \mathcal{J}_{hete} provides an optimization direction consistent with the homogeneous objective \mathcal{J}_{homo} . Specifically, we prove that the gradient of \mathcal{J}_{hete} with respect to the policy parameters is aligned with the gradient of the log-likelihood of the optimal policy.

Assumption D.7. (Importance Sampling Approximation). We assume that the sequence-level importance sampling ratio $s_i(\theta)$ for the learner's self-generated responses remains approximately unity during the gradient update step. That is, we approximate:

$$s_i(\theta) = \left(\frac{\pi_\theta(y_i)}{\pi_{\theta_{old}}(y_i)} \right)^{\frac{1}{|y_i|}} \approx 1 \quad (31)$$

Remark D.8. Unlike standard token-level importance sampling, which suffers from high variance due to the product of probabilities, our method is based on GSPO (Zheng et al., 2025), which employs sequence-level length normalization (geometric mean). This normalization effectively counteracts the cumulative divergence of probability ratios, constraining the value of $s_i(\theta)$ to a stable range centered at 1.0.

Definition D.9. With Assumption D.7, we can only focus on the discrepancy between π_{θ_1} and π_{θ_2} . For succinctness, let $P(y)$ and $Q(y)$ denote the sequence-level likelihood probabilities of a response y generated by the current policy of Agent 1 (π_{θ_1}) and Agent 2 (π_{θ_2}), respectively:

$$P(y) = \prod_{t=1}^{|y|} \pi_{\theta_1}(y_t \mid x, y_{<t}), \quad Q(y) = \prod_{t=1}^{|y|} \pi_{\theta_2}(y_t \mid x, y_{<t}). \quad (32)$$

Recall from Assumption D.7 and Section D.2.1 that the gradient of the homogeneous objective satisfies the alignment condition:

$$\nabla_{\theta_1} \mathcal{J}_{\text{homo}} \approx \mathbb{E}_{y \sim P} \left[\hat{A} \frac{1}{|y|} \nabla_{\theta_1} \log P(y) \right] \quad (33)$$

To prove the effectiveness of $\mathcal{J}_{\text{hete}}$, it suffices to show that $\nabla_{\theta_1} \mathcal{J}_{\text{hete}}$ shares this orientation.

Lemma D.10. For two probability distributions $P(y)$ and $Q(y)$, the following equality holds:

$$\mathbb{E}_{y \sim Q} \left[\frac{P(y)}{Q(y)} f(y) \right] = \mathbb{E}_{y \sim P} [f(y)].$$

Using the importance sampling lemma in Lemma D.10, we present the following theorem regarding the alignment of the heterogeneous gradient.

Theorem D.11. The gradient of the heterogeneous objective $\nabla_{\theta_1} \mathcal{J}_{\text{hete}}$ has a positive angle with the gradient of the homogeneous objective $\nabla_{\theta_1} \mathcal{J}_{\text{homo}}$. That is:

$$\langle \nabla_{\theta_1} \mathcal{J}_{\text{hete}}, \nabla_{\theta_1} \mathcal{J}_{\text{homo}} \rangle > 0. \quad (34)$$

Proof. The heterogeneous objective is defined as an expectation over samples y drawn from Agent 2 ($y \sim Q$):

$$\mathcal{J}_{\text{hete}} = \mathbb{E}_{y \sim Q} \left[\omega^{(2,1)} \text{sg}[s_i^{\text{hete}}]^\alpha \hat{A}_i s_i^{\text{hete}} \right]. \quad (35)$$

We replace the corresponding term in Equation (29) with $P(y)$ and $Q(y)$ defined in Definition D.9, and then apply Lemma D.10:

$$\begin{aligned} \nabla_{\theta_1} \mathcal{J}_{\text{hete}}(\theta_1, \theta_2) &= \mathbb{E}_{y \sim Q} \left[\omega^{(2,1)} \left(\frac{P(y)}{Q(y)} \right)^{\frac{\alpha+1}{|y|}} \cdot \hat{A}_i \frac{1}{|y|} \nabla_{\theta_1} \log P(y) \right] \\ &= \mathbb{E}_{y \sim Q} \left[\left(\frac{P(y)}{Q(y)} \right) \cdot \omega^{(2,1)} \left(\frac{P(y)}{Q(y)} \right)^{\frac{\alpha+1}{|y|}-1} \cdot \hat{A}_i \frac{1}{|y|} \nabla_{\theta_1} \log P(y) \right] \end{aligned} \quad (36)$$

Using the identity $\mathbb{E}_{y \sim Q} \left[\frac{P(y)}{Q(y)} f(y) \right] = \mathbb{E}_{y \sim P} [f(y)]$:

$$\nabla_{\theta_1} \mathcal{J}_{\text{hete}}(\theta_1, \theta_2) = \mathbb{E}_{y \sim Q} \left[\underbrace{\frac{P(y)}{Q(y)} \cdot \omega^{(2,1)} \left(\frac{P(y)}{Q(y)} \right)^{\frac{\alpha+1}{|y|}-1}}_{f(y)} \cdot \hat{A}_i \frac{1}{|y|} \nabla_{\theta_1} \log P(y) \right] \quad (37)$$

$$= \mathbb{E}_{y \sim P} \left[\underbrace{\omega^{(2,1)} \left(\frac{P(y)}{Q(y)} \right)^{\frac{\alpha+1}{|y|}-1}}_{C(y)} \cdot \hat{A}_i \frac{1}{|y|} \nabla_{\theta_1} \log P(y) \right]. \quad (38)$$

For succinctness, let $g(y) = \hat{A}_i \frac{1}{|y|} \nabla_{\theta_1} \log P(y)$:

$$\begin{aligned} \nabla_{\theta_1} \mathcal{J}_{hete}(\theta_1, \theta_2) &= \mathbb{E}_{y \sim P}[C(y) \cdot g(y)] \\ &= \mathbb{E}_{y \sim P}[C(y)] \cdot \mathbb{E}_{y \sim P}[g(y)] + \text{Cov}(C(y), g(y)) \end{aligned} \quad (39)$$

Let define a constant vector \mathbf{v} as:

$$\mathbf{v} := \mathbb{E}_{y \sim P}[g(y)], \quad \nabla_{\theta_1} \mathcal{J}_{homo} \approx \mathbf{v} \quad (40)$$

To prove that the heterogeneous update provides a valid optimization direction, we analyze the inner product between the two gradients. Let $\mathcal{I} = \langle \nabla_{\theta_1} \mathcal{J}_{hete}, \nabla_{\theta_1} \mathcal{J}_{homo} \rangle$.

$$\begin{aligned} \mathcal{I} &\approx \langle \mathbb{E}_{y \sim P}[C(y)] \cdot \mathbf{v} + \text{Cov}(C(y), g(y)), \mathbf{v} \rangle \\ &= \mathbb{E}_{y \sim P}[C(y)] \cdot \langle \mathbf{v}, \mathbf{v} \rangle + \langle \text{Cov}(C(y), g(y)), \mathbf{v} \rangle \\ &= \mathbb{E}_{y \sim P}[C(y)] \cdot \|\mathbf{v}\|^2 + \text{Cov}(C(y), \langle g(y), \mathbf{v} \rangle) \end{aligned} \quad (41)$$

Let $Z(y) = \langle g(y), \mathbf{v} \rangle$ be a scalar random variable representing the alignment between the single-sample gradient $g(y)$ and the expected homogeneous gradient direction \mathbf{v} . Substituting this into Equation (44), we obtain:

$$\mathcal{I} = \mathbb{E}_{y \sim P}[C(y)] \cdot \|\mathbf{v}\|^2 + \text{Cov}(C(y), Z(y)) \quad (42)$$

For the heterogeneous update to provide a valid optimization direction (i.e., $\mathcal{I} > 0$), the weighting coefficient $C(y)$ must satisfy the following condition:

$$\text{Cov}(C(y), Z(y)) > -\mathbb{E}_{y \sim P}[C(y)] \cdot \|\mathbf{v}\|^2 \quad (43)$$

Let $\rho_{C,Z}$ be the correlation coefficient between $C(y)$ and $Z(y)$, and let σ_C, σ_Z denote their respective standard deviations. The condition for positive alignment (Eq. 43) can be rewritten as:

$$\rho_{C,Z} \cdot \sigma_C \cdot \sigma_Z > -\mathbb{E}[C(y)] \cdot \|\mathbf{v}\|^2 \quad (44)$$

It is worth to notion that:

$$C(y) \approx \omega^{(2,1)} \frac{Q(y)}{P(y)}, \quad (45)$$

because that $\alpha + 1$ is far less than $|y|$, therefor $\frac{\alpha+1}{|y|} - 1 \approx -1$.

To guarantee the satisfaction of the condition in Eq. 44, we introduce a mild assumption regarding the collaborative nature of the heterogeneous agents.

Assumption D.12 (Positive Competence Alignment). We assume that the Agent 2 is a competent collaborator, meaning its confidence is positively correlated with the response quality. Mathematically, the correlation coefficient between the weighting coefficient $C(y)$ and the gradient alignment $Z(y)$ is positive:

$$\rho_{C,Z} > 0. \quad (46)$$

Remark D.13 (Physical Interpretation). The assumption $\rho_{C,Z} > 0$ essentially posits that the sampler (Agent 2) acts as a competent collaborator rather than an adversary. A high weight $C(y)$ indicates the sampler's superior confidence relative to the learner, while a high $Z(y)$ indicates a high-quality response. The positive correlation implies that the sampler's confidence is generally aligned with the ground-truth reward signal, thereby facilitating effective knowledge transfer.

The coefficient term (Equation $C(y)$) is strictly positive for all valid trajectories because the capability ratio $\omega^{(2,1)} > 0$, and the probability ratio is non-negative. Thus, the condition in Eq. 44 will be satisfied.

This confirms that the optimization direction of \mathcal{J}_{hete} is consistent with that of \mathcal{J}_{homo} , ensuring that cross-agent responses effectively contribute to the improvement of Agent 1. \square

E. Formulation and Pseudocode of HACPO

To facilitate a precise understanding of HACPO, we present the complete algorithmic formulation and training procedure.

Taking two agents (1 and 2) as an example. The optimization objective for agent 1 consists of two terms: the loss computed from its own samples, $J_{\text{homo}}(\theta)$, and the loss computed from samples of other agents, $J_{\text{hete}}(\theta)$. The final loss is the sum of these two terms. Similarly, agent 2 is updated using a loss function of the same form, but with different values.

$$\mathcal{J}_{\text{homo}}^{(1)} = \frac{1}{G} \sum_{i=1}^G \left[\min \left(s_{t,i}^{(1,1)}, \text{clip} \left(s_{t,i}^{(1,2)} \right) \right) \cdot A_{t,i}^{(1)} \right] \quad (47)$$

$$s_{t,i}^{(1,1)} = \left(\frac{\pi_{\theta}^{(1)}(y_i)}{\pi_{\theta_{\text{old}}}^{(1)}(y_i)} \right)^{\frac{1}{|y_i|}} \quad (48)$$

$$\text{clip}(s_{t,i}^{(1,1)}) = \text{clip}(s_{t,i}^{(1,1)}, 1 - \epsilon_l, 1 + \epsilon_h) \quad (49)$$

$$\mathcal{J}_{\text{hete}}^{(1)} = \frac{1}{G} \sum_{i=1}^G \left[\text{clip} \left(s_{t,i}^{(1,2)} \right) \text{sg} \left(s_{t,i}^{(1,2)} \right)^{\alpha} \omega_t^{(2,1)} \cdot A_{t,i}^{(1)} \right], \quad (50)$$

$$s_{t,i}^{(1,2)} = \left(\frac{\pi_{\theta}^{(1)}(y_i)}{\pi_{\theta_{\text{old}}}^{(2)}(y_i)} \right)^{\frac{1}{|y_i|}} \quad (51)$$

$$\text{clip}(s_{t,i}^{(1,2)}) = \text{clip}(s_{t,i}^{(1,2)}, 1.0 - \delta + k \cdot \delta_{\text{step}}, 1.0) \quad (52)$$

$$\mathcal{J} = \mathcal{J}_{\text{homo}} + \mathcal{J}_{\text{hete}} \quad (53)$$

In the Equation 50, $\tilde{s}_{t,i}^{(1,2)}$ and $\tilde{A}_{t,i}^{(1)}$ are unfolded as mentioned in Section 3.2 and 3.3.

Algorithm 1 Heterogeneous Agent Collaborative Policy Optimization

Require: n initial policy models $\pi_{\text{init}_1}, \pi_{\text{init}_2}, \dots, \pi_{\text{init}_n}$, reward models R , task prompts \mathcal{D} , each prompt has G outputs

```

1: for  $i = 1$  to  $n$  do
2:   policy model  $\pi_{\theta_i} \leftarrow \pi_{\text{init}_i}$ 
3: end for
4: for step = 1 to  $N$  do
5:   Sample a batch  $\mathcal{D}_{\text{batch}}$  from  $\mathcal{D}$ 
6:   for  $i = 1$  to  $n$  do
7:     Update the old policy model  $\pi_{\theta_{i,\text{old}}} \leftarrow \pi_{\theta_i}$ 
8:   end for
9:   for  $i = 1$  to  $n$  do
10:    Sample  $G$  output  $o \sim \pi_{\text{old}_i}(\cdot | q)$  for each question  $q \in \mathcal{D}_{\text{batch}}$ 
11:    Compute rewards  $r_j$  for each output  $o_j$  in the batch
12:    Compute accuracy for the sampling model
13:   end for
14:   for  $i = 1$  to  $n$  do
15:    Compute  $A_{i,o}$  for the response in batch (agent  $i$ )
16:    for mini batch = 1 to  $k$  do
17:      Update the policy model  $\pi_{\theta_i}$  by maximizing the HACPO objective
18:    end for
19:   end for
20: end for

```

Ensure: $\pi_{\theta_i} | i = 1, 2, \dots, n$

F. Additional Experimental Results

Here, we present additional experiments in Table 7, including comparisons between Qwen3-4B-Base + Qwen3-8B-Base, Llama3.2-1B-Instruct + Llama3.2-3B-Instruct, and Qwen3-1.7B-Base + Llama3.2-1B-Instruct.

Table 7. Additional Experimental Results

Model	MATH-500	math	gsm8k	aime2025	AMC23	minerva	olympiad	AVG
Qwen3-4B-Base and Qwen3-8B-Base								
4B-Base	0.61	0.676	0.445	0.1	0.4	0.308	0.347	0.412
4B-Base(GRPO)	0.796	0.788	0.885	0.307	0.475	0.349	0.454	0.579
4B-Base(GSPO)	0.782	0.787	0.877	0.25	0.525	0.368	0.46	0.578
4B-Base(GSPO \times 2)	0.756	0.794	0.873	0.208	0.55	0.382	0.463	0.575
4B-Base(Naive)	0.734	0.712	0.895	0.143	0.55	0.342	0.354	0.526
4b-Base(HACPO)	0.81	0.803	0.904	0.275	0.6	0.364	0.463	0.614
8B-Base	0.647	0.713	0.684	0.033	0.4	0.232	0.375	0.441
8B-Base(GRPO)	0.814	0.812	0.921	0.265	0.575	0.415	0.479	0.612
8b-Base(GSPO)	0.794	0.804	0.923	0.225	0.6	0.426	0.468	0.606
8b-Base(GSPO \times 2)	0.8	0.803	0.92	0.2	0.575	0.404	0.46	0.595
8b-Base(Naive)	0.79	0.783	0.921	0.252	0.5	0.408	0.429	0.583
8B-base(HACPO)	0.828	0.813	0.933	0.323	0.625	0.423	0.467	0.63
Llama3.2-1B-Instruct and Llama3.2-3B-Instruct								
llama3.2-1B	0.176	0.297	0.489	0	0.15	0.052	0.061	0.18
llama3.2-1B(GRPO)	0.35	0.349	0.569	0	0.125	0.008	0.097	0.214
llama3.2-1B(GSPO)	0.356	0.346	0.523	0.021	0.125	0.066	0.088	0.218
llama3.2-1B(GSPO \times 2)	0.352	0.349	0.573	0.07	0.125	0.079	0.103	0.227
llama3.2-1B(Naive)	0.284	0.302	0.45	0.0	0.025	0.066	0.073	0.171
llama3.2-1B(HACPO)	0.35	0.352	0.541	0.022	0.2	0.081	0.085	0.233
llama3.2-3B	0.267	0.441	0.788	0.0	0.2	0.169	0.158	0.289
llama3.2-3B(GRPO)	0.502	0.507	0.814	0.0	0.25	0.199	0.174	0.349
llama3.2-3B(GSPO)	0.512	0.501	0.812	0.054	0.225	0.184	0.17	0.351
llama3.2-3B (GSPO \times 2)	0.488	0.498	0.829	0.0	0.175	0.188	0.159	0.334
llama3.2-3B(Naive)	0.406	0.407	0.734	0.0	0.225	0.177	0.107	0.294
llama3.2-3B(HACPO)	0.522	0.51	0.828	0.067	0.275	0.199	0.188	0.37
Qwen3-1.7B-Base and Llama3.2-1B-Instruct								
qwen3-1.7B	0.5	0.483	0.616	0.033	0.3	0.206	0.229	0.338
qwen3-1.7B(GRPO)	0.682	0.652	0.824	0.16	0.375	0.272	0.298	0.466
qwen3-1.7B(GSPO)	0.648	0.641	0.826	0.148	0.45	0.272	0.287	0.467
qwen3-1.7B(GSPO \times 2)	0.664	0.65	0.829	0.177	0.375	0.265	0.293	0.475
qwen3-1.7B(Naive)	0.59	0.596	0.798	0.105	0.3	0.221	0.241	0.407
qwen3-1.7B(HACPO)	0.676	0.661	0.838	0.22	0.45	0.305	0.32	0.496
llama3.2-1B	0.176	0.297	0.489	0.033	0.15	0.052	0.061	0.18
llama3.2-1B(GRPO)	0.35	0.349	0.569	0	0.125	0.008	0.097	0.214
llama3.2-1B(GSPO)	0.356	0.346	0.523	0.021	0.125	0.066	0.088	0.218
llama3.2-1B(GSPO \times 2)	0.352	0.349	0.573	0.07	0.125	0.079	0.103	0.227
llama3.2-1B(Naive)	0.336	0.337	0.512	0.0	0.125	0.066	0.071	0.214
llama3.2-1B(HACPO)	0.356	0.368	0.533	0.033	0.15	0.066	0.091	0.228