CO-RL: UNSUPERVISED REASONING EMERGES FROM DIVERSE COHORT IN MULTI-AGENT RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning (RL) has emerged as a powerful approach for improving reasoning in language and vision-language models, yet its strongest successes still depend heavily on ground-truth supervision (e.g., verifiable reward). Such annotations are costly to obtain and become increasingly scarce as reasoning capabilities advance beyond what humans can reliably evaluate. Self-rewarding RL reduces this dependence by enabling models to derive reward signals from their own completions. However, training solely on self-generated feedback can reinforce existing biases, reduce response diversity, and ultimately lead to homogenized responses and training collapse. In this work, we show that unsupervised reasoning can emerge through cooperative multi-agent training. We introduce CO-RL, a framework in which multiple decoupled models, sharing no parameters, are simultaneously optimized through RL using rewards derived from their peers. We further show that increasing cohort diversity, through heterogeneous model families, sizes, and rephrased training samples, reduces the correlated errors that drive self-reinforcing feedback loops. This diversity consistently improves reasoning performance and mitigates training collapse. Without any ground-truth labels, CO-RL improves the base models by 3.0–7.2% on average across seven benchmarks and by 1.6–4.4% across four multimodal benchmarks. It also outperforms the prior state-of-the-art multi-agent RL method CoMAS by 4.0% and the self-rewarding vision-language RL method MM-UPT by 2.0%.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has emerged as a powerful approach for improving reasoning in large language models (Lightman et al., 2024; DeepSeek-AI, 2025), yet its strongest successes still depend heavily on ground-truth supervision. Such supervision is costly to obtain and becomes increasingly scarce as target reasoning capabilities approach or surpass what humans can reliably evaluate (Yue et al., 2025). Self-rewarding RL reduces this dependence by deriving rewards from the model's own completions, incorporating signals such as agreement with its majority-vote prediction (Zuo et al., 2025), self-certainty (Zhao et al., 2026), predictive entropy (Prabhudesai et al., 2025), or consistency across paraphrased inputs or moving-average policies (Zhang et al., 2026b). However, these signals remain within a single model's own predictions. Without an external reference, such self-reinforcement can amplify existing biases and suboptimal behaviors, reduce response diversity, and ultimately lead to increasingly homogeneous outputs or even training collapse.

This raises a fundamental question: *how can a model obtain a sufficiently independent learning signal to improve without any ground-truth supervision?* In this work, we show that such a signal can emerge from independently trained models. Since their errors are not perfectly correlated, each model can provide corrective feedback that the other cannot derive from its own generations. We therefore take the learning signal from a separate model, giving each agent *decorrelated supervision*: a target produced by independently updated weights that is less likely to echo its own biases (Blum & Mitchell, 1998; Li et al., 2023).

Building on this insight, we introduce CO-RL, a cooperative multi-agent label-free RL method in which multiple decoupled models, sharing no parameters, are optimized simultaneously using rewards derived from their cohorts. Given an unlabeled prompt, each agent samples multiple completions and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

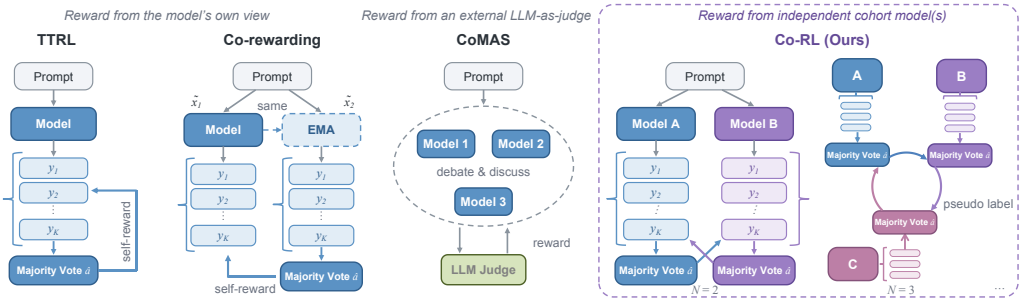


Figure 1: **Comparison of Co-RL with prior label-free RL methods.** TTRL and Co-rewarding derive rewards from self-generated majority-vote responses, while CoMAS evaluates multi-turn interactions using one of its own agents as an LLM judge. In contrast, Co-RL derives rewards from independent peer votes, decoupling the reward signal from the agent being optimized.

aggregates them into a pseudo-label through majority voting (Wang et al., 2023). The completions of one agent are then rewarded against another agent's pseudo-label, and the resulting rewards drive policy optimization with GRPO (Shao et al., 2024). Unlike self-rewarding methods, where the policy and the reward come from the same model, Co-RL draws supervision from an independently updated peer, weakening the direct feedback loop that amplifies a model's own bias. Cohort diversity is essential to Co-RL, as highly similar agents are more likely to make correlated errors, increasing the risk of reinforcing the same incorrect predictions. To promote complementary behaviors, we diversify agents along several dimensions, including model architecture and pretrained weights, model scale, and input formulation (e.g., rephrased prompts (Zhang et al., 2026b)). These sources of heterogeneity expose agents to distinct inductive biases and decision boundaries, reducing error correlation and strengthening the corrective signals available within the cohort.

As illustrated in Figure 1, Co-RL differs fundamentally from both self-rewarding and multi-agent RL approaches. Whereas self-rewarding methods derive rewards from an agent's own predictions, Co-RL obtains supervision from independent peers, decoupling reward generation from the agent being optimized. Unlike prior multi-agent RL methods, it further requires no explicit inter-agent interaction beyond reward construction and no external LLM judge or learned reward model (Xue et al., 2026; Park et al., 2025; Zhang et al., 2026a). This yields a lightweight and symmetric framework in which every agent serves both as a learner and as a source of supervision for its peers.

We benchmark Co-RL across diverse model families in both text-only and multimodal domains. Across seven text-only benchmarks spanning mathematical reasoning, coding, and knowledge-intensive reasoning, Co-RL improves the base models by 3.0–7.2% on average and outperforms the strongest label-free baselines by 0.7–2.0%. The gains extend to multimodal reasoning, where Co-RL improves vision-language models (VLMs) by 1.6–4.4% across four benchmarks. Co-RL also surpasses the prior best multi-agent method such as CoMAS (Xue et al., 2026) by 4.0% and the multimodal self-rewarding method MM-UPT (Wei et al., 2025) by 2.0%.

2 Preliminary

Reinforcement learning for reasoning. Given a prompt $x \sim \mathcal{D}$, an autoregressive language model π_θ generates a response $y = (y_1, \ldots, y_T)$ according to $\pi_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t})$. For reasoning tasks, the response typically contains both an intermediate reasoning trajectory and a final answer, while a scalar reward $r(x, y)$ is commonly assigned at the response level based on answer correctness or another outcome-based criterion.

Among policy optimization algorithms, Group Relative Policy Optimization (GRPO) has been widely adopted because it does not require a separately trained critic model (Shao et al., 2024; DeepSeek-AI, 2025). Given a prompt x, the policy samples a group of K responses $\{y^1, \ldots, y^K\}$ and assigns each response a reward $r^k = r(x, y^k)$. GRPO estimates the advantage of each response by normalizing

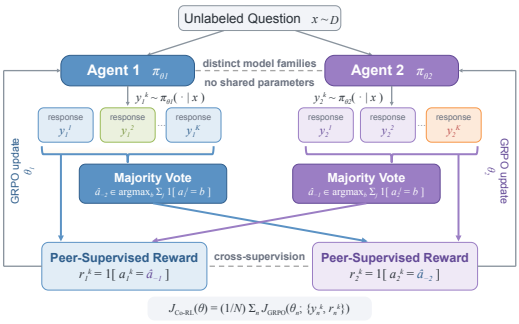


Figure 2: Training procedure of Co-RL with two agents.

Algorithm 1 Co-RL

Require: Agents $\{\pi_{\theta_n}\}_{n=1}^{N}$, unlabeled prompts $\mathcal{D}$
for each training step **do**
  Sample a batch $\mathcal{B} \subset \mathcal{D}$
  for $x \in \mathcal{B}, n = 1, \ldots, N$ **do**
    Sample $y_n^k \sim \pi_{\theta_n}(\cdot \mid x)$, $a_n^k \leftarrow g(y_n^k)$
  end for
  for $x \in \mathcal{B}, n = 1, \ldots, N$ **do**
    $\hat{a}_{-n}(x) \leftarrow$ majority vote of $\{a_{n-1}^k\}_{k=1}^{K}$
    $r_n^k \leftarrow \mathbf{1}[a_n^k = \hat{a}_{-n}(x)]$ for $k = 1, \ldots, K$
  end for
  Update each θ_n with GRPO on $\{y_n^k, r_n^k\}_{k=1}^{K}$
end for

rewards within the group,

$$\hat{A}^k = \frac{r^k - \text{mean}(\{r^j\}_{j=1}^{K})}{\text{std}(\{r^j\}_{j=1}^{K}) + \epsilon}, \tag{1}$$

where ϵ is a small constant for numerical stability. These advantages are used in the standard clipped GRPO objective; Appendix B gives the complete formulation.

Self-rewarding RL. In the absence of external verifiers, reward functions can be constructed directly from a model's own outputs. Given responses $\{y^1, \ldots, y^K\}$ with extracted answers $a^k = g(y^k)$, TTRL (Zuo et al., 2025) assigns $r_{\text{maj}}^k = \mathbf{1}[a^k = \hat{a}_\theta(x)]$, where $\hat{a}_\theta(x) \in \arg\max_b \sum_{j=1}^{K} \mathbf{1}[a^j = b]$ is the policy's majority-vote answer. Intuitor (Zhao et al., 2026) and RENT (Prabhudesai et al., 2025) instead derive rewards from token-level confidence and predictive entropy. Despite their different constructions, these signals originate from the same policy being optimized. Without an external reference, training may reinforce existing biases and suboptimal behaviors. Co-RL replaces this self-generated target with a peer-generated target.

3 Method

3.1 Co-Reinforcement Learning (Co-RL)

In this work, we introduce Co-Reinforcement Learning (Co-RL), a label-free multi-agent reinforcement learning framework in which multiple decoupled agents independently solve unlabeled problems and provide supervision to one another through majority-vote pseudo-rewards. The agents share neither parameters nor gradients, and their optimization is coupled solely through the peer-generated reward signals. Formally, let $x \sim \mathcal{D}$ denote an unlabeled reasoning problem. We consider a cohort of N independent agents parameterized by $\{\pi_{\theta_n}\}_{n=1}^{N}$. The agents may be initialized from the same pretrained model or from models of different families and sizes. For each problem x, agent n independently rolls out a group of K completions $y_n^k \overset{\text{i.i.d.}}{\sim} \pi_{\theta_n}(\cdot \mid x)$, $k = 1, \ldots, K$, and extracts the corresponding final answers as $a_n^k = g(y_n^k)$, with $g(\cdot)$ denoting an answer-extraction function.

For each agent n, Co-RL constructs a supervision target exclusively from the answers generated by one designated peer. Specifically, the peer-derived pseudo-label and the resulting reward for agent n are defined as $\hat{a}_{-n}(x) \in \arg\max_b \sum_{j=1}^{K} \mathbf{1}[a_{n-1}^j = b]$ and $r_n^k = \mathbf{1}[a_n^k = \hat{a}_{-n}(x)]$, where b ranges over the answers produced by agent $n-1$, with the index taken cyclically so that agent 1 is supervised by agent N. Thus, $\hat{a}_{-n}(x)$ represents the majority-vote answer of the peer that supervises agent n. A response receives reward 1 when its extracted answer agrees with the constructed pseudo-label, and reward 0 otherwise. In the two-agent setting, the two agents supervise each other. Crucially, each agent does not contribute to its own supervision target $\hat{a}_{-n}(x)$.

Policy optimization. Each agent is trained with GRPO (Shao et al., 2024), while the proposed design remains optimizer-agnostic and can be incorporated into other RL frameworks. For each agent, the rewards $\{r_n^k\}_{k=1}^{K}$ are normalized within its own rollout group to obtain group-relative advantages.

At each training step, all rollouts and majority-vote pseudo-labels are computed before any policy update. Given the resulting rewards, each policy is then updated independently. Figure 2 and Algorithm 1 give the complete procedure.

3.2 Diverse Cohorts Enable Unsupervised Reasoning

A central design principle of Co-RL is that useful peer supervision should be both *decoupled* from the learner and *complementary* to its own predictions. Decoupling alone prevents an agent from directly constructing rewards from the same predictions it is optimizing, but it does not guarantee that the resulting supervision is informative. If two agents exhibit nearly identical behaviors, their errors can remain highly correlated, causing the cohort to repeatedly reinforce the same incorrect answers. The effectiveness of Co-RL therefore depends not only on separating the learner from its reward source, but also on promoting diversity among the agents that constitute the cohort.

Decoupled supervision. Self-rewarding methods construct supervision from a model's own samples, coupling the training target to the policy being optimized. This creates a self-reinforcing loop in which incorrect predictions can become training targets and be further amplified. In contrast, Co-RL constructs each agent's reward exclusively from its peers, preventing the agent from directly reinforcing its own output. This decoupling breaks the self-confirmation loop without parameter sharing or gradient exchange. As shown in Figure 3(b,c), TTRL becomes unstable and eventually collapses, whereas Co-RL maintains stable pseudo-label quality and accuracy.

Diversity across model families and scales. Decoupling is most effective when peers make complementary rather than correlated errors. We therefore construct heterogeneous cohorts by varying both model family and model scale. Distinct families differ throughout the model-development pipeline, including architecture, tokenization, pretraining data and post-training. Vision-language model families also use distinct vision encoders (Bai et al., 2025; Chen et al., 2024b). Figure 3(d) shows that models from different families overlap less in their errors than models from the same family before training. During training, agents from the same family quickly reach high agreement, while agents from different families keep disagreeing on more problems (Figure 3(a)). The disagreement gives each agent useful corrections and leads to higher accuracy for most of training (Figure 3(c)). Varying the scale lets a smaller model train with a larger one and both still improve (Section 5.2). The weaker agent thus supervises the stronger one as in weak-to-strong supervision (Burns et al., 2024).

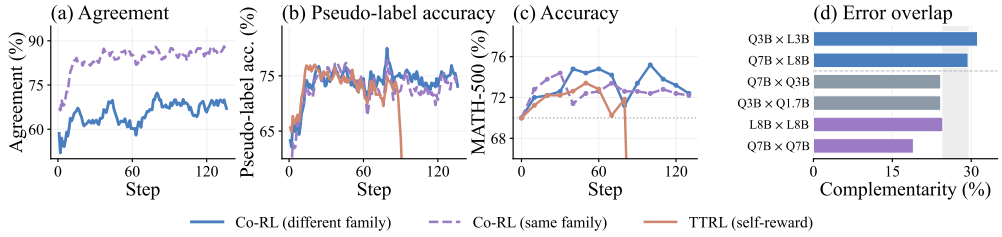


Figure 3: (a) Agreement rate between the majority votes of the two agents; (b) pseudo-label accuracy; (c) evaluation performance; (d) error overlap for different base-model pairs before RL.

Diversity across input formulations. Cohort diversity need not arise solely from model parameters. We further diversify agents by presenting different input formulations of the same underlying problem (Zhang et al., 2026b). For MATH, each training problem is rewritten using DeepSeek-R1 (DeepSeek-AI, 2025); one agent is trained on the original prompt and the other on the rewritten version. These semantically equivalent but stylistically distinct inputs can induce different reasoning trajectories and reduce correlated reliance on the same lexical or structural cues, providing an additional source of complementary supervision. Examples are provided in Appendix E.

4 Theoretical Analysis

In this section, we theoretically characterize the learning dynamics of Co-RL and compare them with self-rewarding GRPO. A key limitation of self-rewarding is that an agent learns from a pseudo-label derived from its own predictions. As a result, when the agent is systematically wrong, its

supervision signal is likely to reinforce the same error rather than correct it. In contrast, CO-RL derives supervision from another agent, allowing one agent's correct prediction to provide a corrective signal for another agent's mistake.

We consider a fixed prompt x and reduce the answer distribution to the correct answer $a^\star$ and an aggregate incorrect answer. Let $p_n = \Pr_{\pi_{\theta_n}}(a = a^\star \mid x)$ denote the probability that agent n assigns to the correct answer and in the single-agent setting we write p. We analyze expected continuous-time updates of the logits $\ell_n = \log(p_n/(1-p_n))$. Each agent draws the same odd number $K \geq 3$ of independent rollouts, and the two rollout groups are conditionally independent given the current policies. We assume a common learning rate $\eta > 0$ for the symmetric two-agent analysis, while the general cross-agent dynamics allow agent-specific rates $\eta_n > 0$. We use normalized group-relative advantages, assign zero update to constant-reward groups and omit clipping and KL regularization.

Training dynamics. The reward-induced dynamics under the two supervision schemes are as follows; their derivation is deferred to Appendix D.1.

Proposition 1 (Training dynamics under self-rewarding and cross-agent supervision)**.** *For self-rewarding, let $C \sim \mathrm{Bin}(K, p)$ count correct responses. Then*

$$\dot{p} = \eta p(1-p)\mathbb{E}_{C\sim\mathrm{Bin}(K,p)}\left[\mathrm{sign}\left(C - \frac{K}{2}\right)\frac{\sqrt{C(K-C)}}{K}\right]. \tag{2}$$

For cross-agent supervision, let $C_n \sim \mathrm{Bin}(K, p_n)$ and let $Z_{-n} = \mathbf{1}[\hat{a}_{-n}(x) = a^\star]$ indicate whether the pseudo-label from agent n's designated peer is correct. Then

$$\dot{p}_n = \eta_n p_n(1-p_n)\mathbb{E}\left[\mathrm{sign}\left(Z_{-n} - \frac{1}{2}\right)\right]\mathbb{E}_{C_n\sim\mathrm{Bin}(K,p_n)}\left[\frac{\sqrt{C_n(K-C_n)}}{K}\right]. \tag{3}$$

Let $V_K(p) = \sum_{c=(K+1)/2}^{K}\binom{K}{c}p^c(1-p)^{K-c}$ be the probability that a majority vote is correct, and define $\phi_K(p) = 2V_K(p) - 1$. Let $q_K(p) = \eta p(1-p)\mathbb{E}_{C\sim\mathrm{Bin}(K,p)}[\sqrt{C(K-C)}/K]$, which is positive for $p \in (0,1)$. For two agents with the same learning rate, Eq. (3) reduces to

$$\dot{p}_A = q_K(p_A)\phi_K(p_B), \qquad \dot{p}_B = q_K(p_B)\phi_K(p_A). \tag{4}$$

Under self-rewarding, the pseudo-label comes from the agent's own rollouts. The update is therefore *self-confirming* and reinforces whichever answer is currently more likely, whether correct or not. In contrast, CO-RL decouples supervision from the optimized agent. Since $q_K(p_A) > 0$, agent A's update direction is determined entirely by the supervision signal $\phi_K(p_B)$ from agent B. The corresponding self-rewarding behavior is summarized below.

Proposition 2 (Self-confirming dynamics)**.** *For odd $K \geq 3$, define*

$$G_K^{\mathrm{self}}(p) = \mathbb{E}_{C\sim\mathrm{Bin}(K,p)}\left[\mathrm{sign}\left(C - \frac{K}{2}\right)\frac{\sqrt{C(K-C)}}{K}\right].$$

For $p \in (0,1)$, $\mathrm{sign}(G_K^{\mathrm{self}}(p)) = \mathrm{sign}(p - 1/2)$. Consequently, self-rewarding sends $p(0) < 1/2$ to $p(t) \to 0$ and $p(0) > 1/2$ to $p(t) \to 1$.

Theorem 1 (CO-RL enlarges the basin of jointly correct convergence)**.** *Consider the symmetric two-agent dynamics in Eq. (4) with interior initialization $(p_A(0), p_B(0)) \in (0,1)^2$. The correct and incorrect consensus states $(1,1)$ and $(0,0)$ are asymptotically stable, while $(1/2, 1/2)$ is a saddle point whose interior separatrix is $p_A + p_B = 1$. Consequently,*

$$(p_A(t), p_B(t)) \longrightarrow \begin{cases} (1,1), & p_A(0) + p_B(0) > 1, \\ (0,0), & p_A(0) + p_B(0) < 1, \\ (1/2, 1/2), & p_A(0) + p_B(0) = 1. \end{cases}$$

In particular, the basins where both *agents converge to the correct answer satisfy*

$$\underbrace{\{p_A > 1/2,\ p_B > 1/2\}}_{\mathcal{B}^+_{\mathrm{self}}} \subsetneq \underbrace{\{p_A + p_B > 1\}}_{\mathcal{B}^+_{\text{Co-RL}}}, \qquad (p_A, p_B) \in (0,1)^2.$$

The proofs of Proposition 2 and Theorem 1 are in Appendices D.2 and D.3.

Takeaway: Co-RL uses complementary strengths to expand the basin of correct convergence. Consider an extreme example where $(p_A, p_B) = (0.9, 0.2)$ on half of the prompts and $(p_A, p_B) = (0.2, 0.9)$ on the other half. Each agent has average accuracy 0.55, but their strengths are perfectly complementary. Under self-rewarding, each agent succeeds only on the half of prompts where $p > 1/2$, yielding a final accuracy of 0.5. In contrast, under Co-RL, $p_A + p_B = 1.1 > 1$ on every prompt, so Theorem 1 predicts that both agents converge to the correct answer on all prompts, achieving accuracy 1.0.

5 Experiments

5.1 Experimental Setup

Data and models. Language models are trained on the level 3 to 5 split of MATH (Hendrycks et al., 2021b), following MARTI (Zhang et al., 2026a). We train Qwen2.5-3B with Llama-3.2-3B-Instruct and Qwen2.5-7B with Llama-3.1-8B-Instruct (Yang et al., 2024; Grattafiori et al., 2024), and add Qwen3-1.7B (Yang et al., 2025a), Phi-4-mini-Instruct (Abouelenin et al., 2025) or Granite-3.3-2B-Instruct (Granite Team, IBM, 2025) as a third agent in the three-agent experiments. Vision-language models are trained on MMR1-Math (Leng et al., 2025), the training data of MM-UPT (Wei et al., 2025), and on multimodal-open-r1 (LMMs-Lab, 2025) as a second dataset. We train Qwen2.5-VL-3B with InternVL3.5-2B and Qwen2.5-VL-7B with InternVL3.5-8B (Bai et al., 2025; Chen et al., 2024b).

Baselines. On language models, we compare our method against previous label-free self-rewarding methods, including TTRL (Zuo et al., 2025), Intuitor (Zhao et al., 2026), RENT (Prabhudesai et al., 2025) and Co-rewarding-II (Zhang et al., 2026b). On vision-language models, we compare with the recent self-rewarding method MM-UPT (Wei et al., 2025), which adopts the TTRL reward formulation. For multi-agent RL baselines, we compare against MAPoRL (Park et al., 2025) and CoMAS (Xue et al., 2026). GRPO trained with ground-truth rewards (GT-Reward) serves as a supervised reference (Shao et al., 2024).

Training and evaluation. Language models follow the training recipe of Co-rewarding (Zhang et al., 2026b) with $K = 12$ responses per prompt, and vision-language models follow R1-V (Chen et al., 2025) with $K = 8$. Language models are evaluated on seven benchmarks in mathematics (GSM8K (Cobbe et al., 2021), MATH-500 (Lightman et al., 2024), AMC (Project-Numina, 2024)), code (HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), LiveCodeBench (Jain et al., 2025)) and science (GPQA (Rein et al., 2024)). Vision-language models are evaluated on MathVision (Wang et al., 2024a), MathVerse (Zhang et al., 2024a), MathVista (Lu et al., 2024) and We-Math (Qiao et al., 2025). All experiments run on a single node of eight H100 GPUs. Appendix F gives the full hyperparameters, evaluation settings and results.

5.2 Results

Co-RL Outperforms All Self-Rewarding Methods on Language Models. Table 1 reports Co-RL+, which trains two models from different families together and gives one of them rephrased questions. Without labels, Co-RL+ improves the base models by 3.0–7.2%. It beats the strongest self-rewarding baseline on every model by up to 2.0% and even surpasses GT-Reward on three of the four models. Although trained only on MATH, it also leads on LiveCodeBench for three models and raises GPQA on Qwen2.5-7B by 8.1% over the strongest baseline. Figure 4 decouples the two agents step by step. Moving from the same family to different families and then adding rephrased questions raises the average accuracy over the four models from 48.4% to 49.0% and 49.5%.

Co-RL Outperforms Multi-Agent RL. Prior state-of-the-art multi-agent RL methods typically rely on an additional judging mechanism to construct rewards, such as an LLM judge in CoMAS or a learned reward model in MAPoRL. For a fair comparison, we use Qwen2.5-3B-Instruct following the same experimental setting as CoMAS. We report the CoMAS results as presented in the original paper. As shown in Table 2, Co-RL surpasses CoMAS by 4.0% on average and by 12.8% on MATH-500. It leads on five of the seven benchmarks while using half as many agents and no judge.

Co-RL Transfers to Vision-Language Models. Vision-language model families differ in both their visual encoders and language backbones, so they test Co-RL under stronger heterogeneity. We train

Table 1: Full performance across seven benchmarks for 3B, 7B and 8B models (%). For each benchmark, the best label-free result is shown in **bold** and the second best is underlined, with ties sharing the marking. Base and GT-Reward are excluded from the ranking. CO-RL+ uses models from different families and trains one of the two agents on rephrased questions. Qwen2.5-3B is trained with Llama-3.2-3B-Instruct and Qwen2.5-7B is trained with Llama-3.1-8B-Instruct.

Method	GSM8K	MATH500	AMC	HEval	GPQA	MBPP	LCB	Avg
				Qwen2.5-3B				
Base	71.4	55.8	25.9	56.1	21.2	50.8	13.7	42.1
GT-Reward	76.2	64.6	36.1	65.2	20.7	54.4	14.5	47.4
TTRL	80.4	66.4	31.3	**63.4**	22.2	51.8	15.9	47.3
RENT	75.6	62.8	31.3	59.2	18.2	52.4	14.5	44.9
Intuitor	74.9	64.2	26.5	59.8	**27.3**	50.4	16.4	45.6
Co-rewarding-II	75.5	63.4	30.1	61.0	24.8	53.2	11.0	45.6
CO-RL+	**81.0**	**66.6**	**36.1**	62.8	25.8	**55.6**	**17.2**	**49.3**
				Llama-3.2-3B-Instruct				
Base	73.6	43.8	18.1	51.2	21.2	50.8	12.0	38.7
GT-Reward	78.8	53.8	25.3	60.4	20.7	50.2	12.1	43.0
TTRL	77.9	50.2	26.5	**59.2**	**24.8**	51.2	12.0	43.1
RENT	75.4	45.2	12.0	**59.2**	17.7	49.4	11.5	38.6
Intuitor	75.8	40.8	21.7	54.3	21.7	**51.4**	12.0	39.7
Co-rewarding-II	75.4	53.4	24.1	54.9	23.7	49.2	**12.1**	41.8
CO-RL+	**78.4**	**55.2**	**30.1**	**59.2**	22.2	50.4	12.0	**43.9**
				Qwen2.5-7B				
Base	82.9	70.0	39.8	73.4	18.7	62.8	21.1	52.7
GT-Reward	84.8	77.6	49.4	80.7	23.7	64.4	25.5	58.0
TTRL	80.6	74.8	39.8	75.6	25.8	**65.4**	23.9	55.1
RENT	78.8	**75.4**	**47.0**	75.4	29.8	61.6	26.2	56.3
Intuitor	**82.9**	**75.4**	41.0	78.9	28.3	64.0	24.8	56.5
Co-rewarding-II	81.9	72.6	43.4	**79.9**	26.8	64.0	25.9	56.4
CO-RL+	80.2	74.4	38.6	79.7	**37.9**	63.2	**26.6**	**57.2**
				Llama-3.1-8B-Instruct				
Base	82.9	49.6	18.1	65.2	22.2	58.4	16.8	44.7
GT-Reward	82.7	53.2	25.3	64.0	30.3	59.2	15.2	47.1
TTRL	83.9	51.0	**27.7**	64.6	21.2	58.2	16.3	46.1
RENT	79.5	48.2	21.7	**67.7**	19.7	**60.0**	16.0	44.7
Intuitor	79.7	45.8	21.7	65.8	26.8	58.0	16.1	44.8
Co-rewarding-II	84.7	52.0	24.1	67.1	22.2	59.8	16.5	46.6
CO-RL+	**85.4**	**55.6**	26.5	64.6	**27.3**	57.2	**17.1**	**47.7**

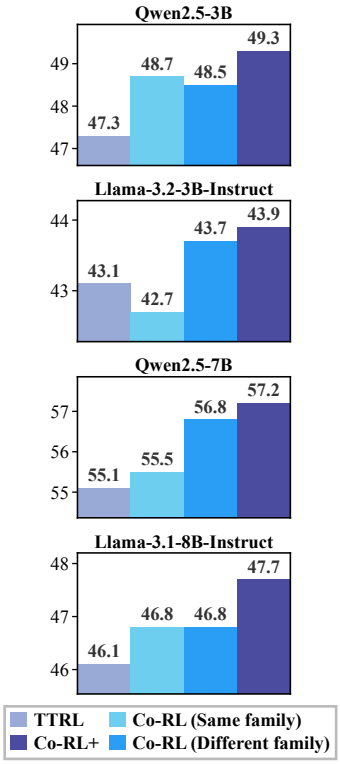


Figure 4: Seven-benchmark average (%) of TTRL and the three CO-RL variants.

Qwen2.5-VL-3B with InternVL3.5-2B and Qwen2.5-VL-7B with InternVL3.5-8B on each of the two multimodal datasets (Appendix F.2). Without labels, CO-RL improves the base models by up to 4.4%. It outperforms TTRL in six of the eight runs and its largest margin is 6.5% on Qwen2.5-VL-7B. At 7B and 8B, it matches or exceeds GT-Reward in all four settings.

CO-RL Outperforms MM-UPT. MM-UPT is a recent label-free RL method for vision-language models. We follow its setting and train Qwen2.5-VL-7B on the same MMR1 data with its evaluation protocol. Table 3 also lists four self-improvement baselines. SRLM (Yuan et al., 2024) rewards its own responses as a judge, LMSI (Huang et al., 2023) fine-tunes on its majority-vote answers, and Genixer (Zhao et al., 2024) and STIC (Deng et al., 2024) train on self-generated data. CO-RL improves the base model by 5.7% and leads all label-free methods on every benchmark. It surpasses MM-UPT by 2.0% on average and by 3.8% on MathVerse. It also exceeds GRPO with ground-truth labels on average, and it needs only about a quarter of the MM-UPT training budget.

CO-RL with Different Sizes and Weak-to-Strong Supervision. We train Qwen2.5-7B together with the smaller Qwen2.5-3B, without labels. As shown in Figure 5(b), both models surpass their TTRL counterparts, and Qwen2.5-3B rises from 42.1% to 47.8%. The weaker model therefore does not hold back the stronger one and improves as well. Llama-3.1-8B-Instruct is larger than Qwen2.5-3B but performs at a similar level before training (44.7% against 42.1%), and it comes from a different family. Figure 5(a) reports the share of the gain from Base to GT-Reward that Qwen2.5-7B recovers (Hase et al., 2024). Training with Llama-3.1-8B-Instruct raises this share from 64% to 77%, while TTRL recovers only 46%. A model from a different family thus supervises better than a same-family one of similar strength, as their errors overlap less (Appendix C).

Table 2: Comparison under the CoMAS setting (%). All methods train on the same prompt mixture and are evaluated following the CoMAS protocol. The CO-RL row uses the Different family setting.

Method	GSM8K	MATH500	HEval	MBPP	MMLU	GPQA	SciBench	Avg
Base	85.40	55.00	73.78	55.80	63.20	28.79	36.47	56.92
MAPoRL	85.80	55.40	75.61	57.00	63.20	**31.47**	**39.08**	58.22
TTRL	88.20	56.80	73.78	59.00	63.80	27.23	38.48	58.18
CoMAS	87.20	55.80	77.44	59.20	65.60	29.69	37.68	58.94
CO-RL	**89.50**	**68.60**	**82.32**	**68.00**	**65.80**	29.69	36.87	**62.97**

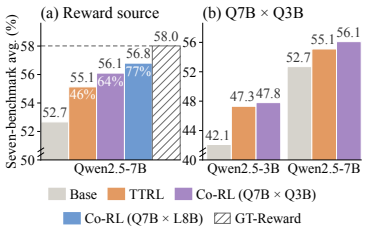


Figure 5: Weak-to-strong supervision (%).

CO-RL with Three Agents. We scale CO-RL to three agents and to model families beyond Qwen and Llama. The agents form a ring in which each agent is rewarded by the majority vote of the previous one (Section 3.1). We train Qwen2.5-3B and Llama-3.2-3B-Instruct together with a third model, which is Qwen3-1.7B, Phi-4-mini-Instruct or Granite-3.3-2B-Instruct. CO-RL matches or exceeds GT-Reward on all five models and beats TTRL on four. On Granite-3.3-2B-Instruct, TTRL collapses because the model repeats one wrong answer in every rollout and its reward saturates. CO-RL therefore scales to three agents and to new model families. Full results in Appendix F.1.

5.3 Ablation Study and Analysis

Effect of the Three Diversity Designs. We next test whether the three designs in Section 3.2 make the agents diverse and whether this diversity improves accuracy. Starting from TTRL, CO-RL (Same family) introduces decoupled supervision, CO-RL (Different family) also uses a model from another family, and CO-RL+ also rephrases the input of one agent. We measure diversity by how often the majority votes of the two agents agree (Li et al., 2023).

Each design induces diversity. Each design lowers the agreement on all four models (Figure 6), from 100% for TTRL by construction to 79.8%, 68.4% and 60.4% at the end of training on average. The effect of the family exists even before training, since exactly one of two models from different families is correct on 29–31% of the problems, against 19–25% for two identical models (Figure 3(d)).

Diversity improves performance. Accuracy rises with each design, from 47.9% for TTRL to 49.5% for CO-RL+ on average (Figure 4). Across the twelve runs, lower agreement also goes with a larger gain over TTRL (Figure 6(b)), with a repeated-measures correlation of $r = -0.91$ ($p = 0.001$) (Bakdash & Marusich, 2017). This gain can only come from the problems on which the two agents disagree. On the other problems, the target from the peer is the agent's own answer, so it cannot correct a mistake. The three designs are therefore effective, because each of them adds such disagreement.

Training Dynamics and Stability. Figure 8(d)–(f) tracks training on Qwen2.5-7B, where every self-rewarding method degrades. RENT and Intuitor lose reward variation and inflate their completions, and TTRL collapses late in training while CO-RL stays stable. These methods reward the model with its own outputs, so the model can raise its reward without answering correctly. TTRL thus reinforces its own wrong votes (Figure 8(b)). In CO-RL, Qwen2.5-7B and Llama-3.1-8B-Instruct supervise each other, so neither can raise its reward by changing only itself. The reward therefore keeps its

Table 3: Comparison under the MM-UPT protocol (%). Results for prior methods are reported from Wei et al. (2025). Base, SFT and GRPO are excluded from the ranking. All CO-RL rows use the Different family setting.

Method	MathVision	MathVerse	MathVista	We-Math	Avg
Base	24.87	43.83	66.30	62.87	49.47
GRPO	29.01	45.03	71.40	67.24	53.17
SFT	26.45	43.53	63.30	64.20	49.37
SRLM	25.33	45.08	67.00	64.66	50.52
LMSI	24.83	43.76	64.90	66.38	49.97
Genixer	23.68	43.30	65.50	64.66	49.29
STIC	23.78	42.72	66.10	63.74	49.09
MM-UPT	26.15	44.87	72.90	68.74	53.17
CO-RL	**28.06**	**48.68**	**73.30**	**70.80**	**55.21**

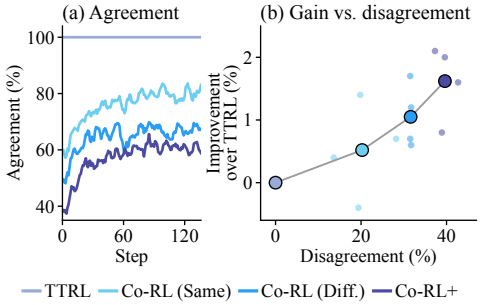


Figure 6: (a) Agreement during training of Qwen2.5-3B. (b) Improvement over TTRL against the final disagreement.

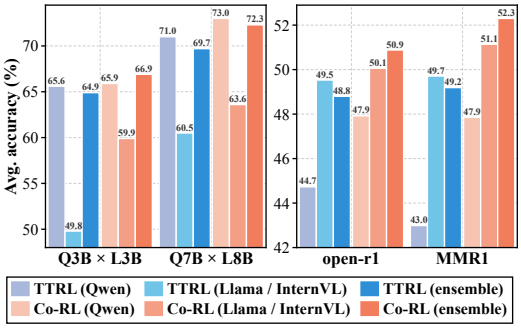


Figure 7: Average accuracy (%) of single models and their ensembles. The left panel shows Qwen2.5-3B with Llama-3.2-3B-Instruct and Qwen2.5-7B with Llama-3.1-8B-Instruct. The right panel shows Qwen2.5-VL-3B with InternVL3.5-2B trained on open-r1 and MMR1.

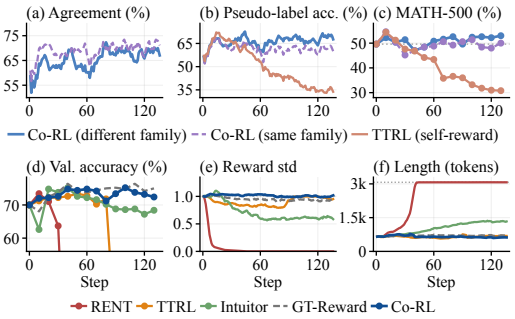


Figure 8: (a)–(c) Llama-3.1-8B-Instruct trained with Qwen2.5-7B (different family), with a second Llama-3.1-8B-Instruct (same family) or alone with TTRL. (d)–(f) Training dynamics of Qwen2.5-7B. Reward standard deviation is normalized to its value at the first step.

variation (Figure 8(e)), and the pseudo-labels they exchange stay accurate (Figure 8(b)). By the end of training, Co-RL leads TTRL on MATH-500 by about 22% on both models.

Fair Comparison under the Same Budgets. Co-RL trains two models, whereas TTRL trains one. Comparing Co-RL with a single TTRL model is unfair, so we train both base models with TTRL to match the training budget. At test time, both methods vote over eight rollouts per question. A single model uses eight of its own rollouts and an ensemble pools four from each model. As shown in Figure 7, Co-RL outperforms TTRL for every model and every ensemble, and its ensemble leads the TTRL ensemble by 2.0–3.1%. The gains of Co-RL therefore come from cross-agent supervision rather than from the extra model or from ensembling. Appendix F.4 reports the full results.

6 Related Work

Self-rewarding RL removes the need for labels by deriving rewards from the model's own outputs, such as its majority vote (Zuo et al., 2025), self-certainty (Zhao et al., 2026) or predictive entropy (Prabhudesai et al., 2025), and has been extended to vision-language models (Wei et al., 2025). These rewards come from the policy itself and can reinforce its own mistakes. Co-training (Blum & Mitchell, 1998) instead lets a second learner supervise the first, and it works best when the two learners make different errors (Li et al., 2023). Co-rewarding (Zhang et al., 2026b) brings this idea to label-free RL but builds both views from the same policy, through paraphrased questions or a slowly updated copy. Multi-agent RL methods train separate models together but rely on multi-turn interaction and a judge or learned reward, such as the LLM judge of CoMAS (Xue et al., 2026) or the reward model of MAPoRL (Park et al., 2025). Co-RL rewards each separately optimized model with the majority vote of its peer and needs no judge. Appendix A discusses further related work.

7 Conclusion

In this work, we introduced Co-RL, a label-free multi-agent RL framework for reasoning tasks. In our framework, multiple agents learn from rewards constructed from their peers' predictions rather than ground-truth labels or external judges. Across text-only and multimodal reasoning benchmarks, Co-RL consistently improves diverse LLMs and VLMs, outperforming prior self-rewarding and multi-agent RL approaches and, in many settings, matching or surpassing training with ground-truth rewards. Our theoretical analysis shows that cross-agent supervision expands the set of initial conditions that converge to the correct solution, allowing Co-RL to correct errors that self-rewarding RL would otherwise reinforce. An important direction for future work is to understand how the number, diversity, and interaction topology of agents shape cross-agent learning, and to develop adaptive supervision mechanisms that more effectively exploit complementary expertise.

AI USE STATEMENT

In this work, we used generative AI tools solely for language editing to improve the grammar, clarity, and readability of the manuscript. All AI-assisted edits were carefully reviewed to ensure that they preserved the intended meaning and technical accuracy. We take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

All authors have read and adhere to the ICLR Code of Ethics. This work trains language and vision-language models on publicly available datasets and evaluates them on public benchmarks. It involves no human subjects, no personal data and no new data collection. All models and datasets are used under their original licenses. We are not aware of ethical risks beyond those of the base models themselves.

REPRODUCIBILITY STATEMENT

Section 5 and Appendix F specify the models, training data, hyperparameters, evaluation protocols and training budgets of all experiments. The assumptions of the theoretical analysis are stated in Section 4, and Appendices D.1, D.2 and D.3 give complete proofs of Propositions 1 and 2 and Theorem 1. Appendix E describes how the rephrased training problems are generated and shows examples. The code is included in the supplementary material, and we will release it together with the rephrased data upon publication.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-Mini technical report: Compact yet powerful multimodal language models via mixture-of-LoRAs. *arXiv preprint arXiv:2503.01743*, 2025.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Jonathan Z. Bakdash and Laura R. Marusich. Repeated measures correlation. *Frontiers in Psychology*, 8:456, 2017. doi: 10.3389/fpsyg.2017.00456.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Conference on Computational Learning Theory (COLT)*, pp. 92–100, 1998.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4971–5012. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/burns24b.html.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9630–9640, 2021.

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7066–7085, 2024a.

Liang Chen, Lei Li, Haozhe Zhao, Yifan Song, and Vinci. R1-V: Reinforcing super generalization ability in vision-language models with less than $3. https://github.com/Deep-Agent/R1-V, 2025. Accessed: 2026-08-17.

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pp. 1597–1607, 2020.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24185–24198, 2024b.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *International Conference on Machine Learning (ICML)*, 2024c.

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

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yihe Deng, Pan Lu, Fan Yin, Ziniu Hu, Sheng Shen, Quanquan Gu, James Zou, Kai-Wei Chang, and Wei Wang. Enhancing large vision language models with self-training on image comprehension. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 131369–131397, 2024.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning (ICML)*, 2024.

Wenkai Fang, Shunyu Liu, Yang Zhou, Kongcheng Zhang, Tongya Zheng, Kaixuan Chen, Mingli Song, and Dacheng Tao. SeRL: Self-play reinforcement learning for large language models with limited data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yicheng Feng, Yijiang Li, Wanpeng Zhang, Sipeng Zheng, Hao Luo, Zihao Yue, and Zongqing Lu. VideoOrion: Tokenizing object dynamics in videos. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20401–20412, 2025.

Granite Team, IBM. Granite 3.3 language models. https://huggingface.co/ibm-granite/granite-3.3-2b-instruct, 2025. Released April 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 21271–21284, 2020.

Peter Hase, Mohit Bansal, Peter Clark, and Sarah Wiegreffe. The unreasonable effectiveness of easy training data for hard tasks. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7002–7024, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.378. URL https://aclanthology.org/2024.acl-long.378/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021a.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2021b.

Shengding Hu, Yuge Tu, Xu Han, et al. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=3X2L2TFr0f.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-Zero: Self-evolving reasoning LLM from zero data. In *International Conference on Learning Representations (ICLR)*, 2026a.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1051–1068, 2023.

Wenxuan Huang, Bohan Jia, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. Vision-R1: Incentivizing reasoning capability in multimodal large language models. In *International Conference on Learning Representations (ICLR)*, 2026b.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations (ICLR)*, 2025.

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik S. Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae W. Park. MDAgents: An adaptive collaboration of LLMs for medical decision-making. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 7, 1994.

Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.

Sicong Leng, Jing Wang, Jiaxi Li, Hao Zhang, Zhiqiang Hu, Boqiang Zhang, Yuming Jiang, Hang Zhang, Xin Li, Lidong Bing, et al. MMR1: Enhancing multimodal reasoning with variance-aware sampling and open resources. *arXiv preprint arXiv:2509.21268*, 2025.

Pengyi Li, Matvey Skripkin, Alexander Zubrey, Andrey Kuznetsov, and Ivan Oseledets. Confidence is all you need: Few-shot RL fine-tuning of language models. *arXiv preprint arXiv:2506.06395*, 2025.

Yijiang Li, Xinjiang Wang, Lihe Yang, Litong Feng, Wayne Zhang, and Ying Gao. Diverse cotraining makes strong semi-supervised segmentor. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Junwei Liao, Muning Wen, Jun Wang, and Weinan Zhang. MARFT: Multi-agent reinforcement fine-tuning. *arXiv preprint arXiv:2504.16129*, 2025.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Bo Liu, Simon Yu, Zichen Liu, Leon Guertler, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, et al. SPIRAL: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-RFT: Visual reinforcement fine-tuning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2034–2044, 2025.

LMMs-Lab. Multimodal open R1. https://huggingface.co/datasets/lmms-lab/multimodal-open-r1-8k-verified, 2025. Accessed: 2026-08-17.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. General-Reasoner: Advancing LLM reasoning across all domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, et al. MM-Eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.

Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman E. Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. MAPoRL: Multi-agent post-co-training for collaborative large language models with reinforcement learning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. LMM-R1: Empowering 3B LMMs with strong reasoning abilities through two-stage rule-based RL. *arXiv preprint arXiv:2503.07536*, 2025.

Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*, 2025.

Project-Numina. AIMO validation AMC. https://huggingface.co/datasets/AI-MO/aimo-validation-amc, 2024. Accessed: 2026-08-17.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma Gongque, Shanglin Lei, Yifan Zhang, et al. We-Math: Does your large multimodal model achieve human-like mathematical reasoning? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *Conference on Language Modeling (COLM)*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. VLM-R1: A stable and generalizable R1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. Corex: Pushing the boundaries of complex reasoning through multi-model collaboration. In *Conference on Language Modeling (COLM)*, 2024.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with MATH-Vision dataset. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024a.

Peiyu Wang, Yichen Wei, Yi Peng, Xiaokun Wang, Weijie Qiu, Wei Shen, Tianyidan Xie, Jiangbo Pei, Jianhao Zhang, Yunzhuo Hao, et al. Skywork R1V2: Multimodal hybrid reinforcement learning for reasoning. *arXiv preprint arXiv:2504.16656*, 2025.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. SciBench: Evaluating college-level scientific problem-solving abilities of large language models. In *International Conference on Machine Learning (ICML)*, 2024b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Lai Wei, Yuting Li, Chen Wang, Yue Wang, Linghe Kong, Weiran Huang, and Lichao Sun. First SFT, second RL, third UPT: Continual improving multi-modal LLM reasoning via unsupervised post-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Wei Xiong, Hanning Zhang, Chenlu Ye, Lichang Chen, Nan Jiang, and Tong Zhang. Self-rewarding correction for mathematical reasoning. *arXiv preprint arXiv:2502.19613*, 2025.

Fangzhi Xu, Hang Yan, Chang Ma, Haiteng Zhao, Qiushi Sun, Kanzhi Cheng, Junxian He, Jun Liu, and Zhiyong Wu. Genius: A generalizable and purely unsupervised self-training framework for advanced reasoning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 13153–13167, 2025a.

Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. KodCode: A diverse, challenging, and verifiable synthetic dataset for coding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6980–7008, 2025b.

Xiangyuan Xue, Yifan Zhou, Guibin Zhang, Zaibin Zhang, Yijiang Li, Chen Zhang, Zhenfei Yin, Philip Torr, Wanli Ouyang, and Lei Bai. CoMAS: Co-evolving multi-agent systems via interaction rewards. In *International Conference on Learning Representations (ICLR)*, 2026.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, et al. R1-OneVision: Advancing generalized multimodal reasoning through cross-modal formalization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2376–2385, 2025b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Kaiyan Zhang, Kai Tian, Runze Liu, Sihang Zeng, Xuekai Zhu, Guoli Jia, Yuchen Fan, Xingtai Lv, Yuxin Zuo, Che Jiang, et al. MARTI: A framework for multi-agent LLM systems reinforced training and inference. In *International Conference on Learning Representations (ICLR)*, 2026a.

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised LLM reasoning incentivization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? In *European Conference on Computer Vision (ECCV)*, pp. 169–186, 2024a.

Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4320–4328, 2018.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö. Arık. Chain of agents: Large language models collaborating on long-context tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

Zizhuo Zhang, Jianing Zhu, Xinmu Ge, Zihua Zhao, Zhanke Zhou, Xuan Li, Xiao Feng, Jiangchao Yao, and Bo Han. Co-rewarding: Stable self-supervised RL for eliciting reasoning in large language models. In *International Conference on Learning Representations (ICLR)*, 2026b.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Henry Hengyuan Zhao, Pan Zhou, and Mike Zheng Shou. Genixer: Empowering multimodal large language model as a powerful data generator. In *European Conference on Computer Vision (ECCV)*, pp. 129–147, 2024.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. In *International Conference on Learning Representations (ICLR)*, 2026.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, et al. TTRL: Test-time reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

A Extended Related Work

Self-rewarding RL. While RLVR effectively improves LLM reasoning (Shao et al., 2024; DeepSeek-AI, 2025; Yu et al., 2025), its reliance on high-quality ground-truth labels remains a major bottleneck (Yue et al., 2025). Recent work therefore explores self-rewarding mechanisms that learn from unlabeled data. One prominent direction derives reward signals solely from a model's own behavior, such as majority voting (Wang et al., 2023), self-consistency (Zuo et al., 2025), self-certainty (Zhao et al., 2026; Li et al., 2025), or predictive entropy (Prabhudesai et al., 2025; Zhang et al., 2025). Because these methods rely on a single model's own view, confident errors may be reinforced without corrective feedback. This line extends earlier work on self-rewarding models (Yuan et al., 2024) and self-play supervision (Chen et al., 2024c), and now includes unsupervised self-training (Xu et al., 2025a; Fang et al., 2025), self-correction based on a model's own judgments (Xiong et al., 2025), and zero-data self-evolution, where one or more models generate their own curricula (Zhao et al., 2025; Huang et al., 2026a; Liu et al., 2026).

Co-training and cross-view supervision. Using a second view to supervise a learner is a longstanding idea. Co-training shows that two conditionally independent views can teach each other (Blum & Mitchell, 1998), while subsequent analysis demonstrates that highly similar views tend to reinforce the same errors: the more alike the two views are, the more they simply agree on the same mistakes (Li et al., 2023). Related principles underlie self-supervised representation learning, where augmentations, momentum targets, or stop-gradient operations prevent collapse (Chen et al., 2020; Grill et al., 2020; Caron et al., 2021), and deep mutual learning, where peer networks provide reciprocal supervision (Zhang et al., 2018). Co-rewarding (Zhang et al., 2026b) derives rewards from paraphrased questions or a slowly updated model copy. Co-RL instead uses separately optimized peers and directly studies the role of cross-family diversity.

Multi-agent RL. Several methods improve reasoning by enabling interactions among multiple models. At inference time, multi-agent debate and round-table consensus have models critique and revise one another (Du et al., 2024; Liang et al., 2024; Chen et al., 2024a; Sun et al., 2024), and general orchestration frameworks compose such agents into pipelines (Wu et al., 2023; Zhang et al., 2024b; Kim et al., 2024). A more recent line trains the agents: CoMAS turns interactions scored by an LLM judge into rewards (Xue et al., 2026), while MAPoRL and MARFT co-train agents against a learned or shared reward (Park et al., 2025; Liao et al., 2025). Co-RL couples agents through peer votes, requiring neither multi-turn communication nor an external reward judge.

RLVR for multimodal reasoning. RLVR has rapidly expanded to vision-language models, with R1-style rule-based rewards underpinning a broad family of methods (Huang et al., 2026b; Shen et al., 2025; Feng et al., 2025; Liu et al., 2025; Meng et al., 2025; Peng et al., 2025; Yang et al., 2025b; Wang et al., 2025). Existing work largely assumes verifiable labels, while label-free multimodal RL remains underexplored and relies on a single model's own signals (Wei et al., 2025). Cross-view supervision is especially promising here because VLM families combine different vision encoders and language backbones.

B Additional Optimization and Training Details

B.1 Complete GRPO Objective

Section 2 defines the group-relative advantage $\hat{A}^k$. Let the token-level importance ratio be

$$\rho_{k,t}(\theta) = \frac{\pi_\theta(y_t^k \mid x, y_{<t}^k)}{\pi_{\theta_{\text{old}}}(y_t^k \mid x, y_{<t}^k)}. \tag{5}$$

Writing the clipped surrogate as $L_{k,t}(\theta) = \min\{\rho_{k,t}(\theta)\hat{A}^k, \text{clip}(\rho_{k,t}(\theta), 1-\delta, 1+\delta)\hat{A}^k\}$, the objective is

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\frac{1}{|y^k|}\sum_{t=1}^{|y^k|}\left(L_{k,t}(\theta) - \beta\mathcal{D}_{\text{KL}}^{k,t}\right)\right]. \tag{6}$$

Here δ is the clipping threshold and β weights deviation from the reference policy. In Co-RL, each agent uses its own responses, peer-generated rewards, old policy, and reference policy in this objective. In Co-RL+, the agent trained on rephrased questions receives the rewritten question as its input x_n. The cohort objective averages these independently optimized surrogates with fixed peer targets,

$$\max_{\boldsymbol{\theta}} \mathcal{J}_{\text{Co-RL}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{J}_{\text{GRPO}}\left(\theta_n; \{y_n^k, r_n^k\}_{k=1}^K\right), \tag{7}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)$.

B.2 Additional Self-Rewarding Objectives

Section 2 gives the majority-vote reward used by TTRL. Intuitor and RENT instead use the following token-level predictive-distribution rewards:

$$r^k_{\text{conf}} = \frac{1}{|y^k|}\sum_{t=1}^{|y^k|} D_{\text{KL}}\big(\mathbf{U} \,\|\, \pi_\theta(\cdot \mid x, y^k_{<t})\big)\,, \qquad \text{Intuitor (Zhao et al., 2026),} \tag{8}$$

$$r^k_{\text{ent}} = -\frac{1}{|y^k|}\sum_{t=1}^{|y^k|} \mathcal{H}\big(\pi_\theta(\cdot \mid x, y^k_{<t})\big)\,, \qquad \text{RENT (Prabhudesai et al., 2025).} \tag{9}$$

Here $\mathbf{U}$ is uniform over the vocabulary and $\mathcal{H}$ is entropy. Each produces scalar rewards that can be normalized into advantages for Eq. (6). Their differing training behavior is evaluated in Section 5.3 and Appendix F.3.

C Pre-training Error Decoupling

Section 3.2 argues that a peer helps when its errors do not overlap with the agent's own and Figure 3(d) previews this overlap for three kinds of model pairs. This appendix reports the full measurement. All numbers are computed on base checkpoints before any RL, so they describe the pretrained models themselves rather than anything our training produces.

Setup and metrics. We score base checkpoints on 500 MATH problems at levels 3 to 5, zero-shot and single-sample at T=0.8, with rule-based extraction and equivalence checking. For a pair of models, every problem lands in one of the four cells of Table 4, and the four diversity measures (Kuncheva & Whitaker, 2003) are counts over these cells. Complementarity c is the share of problems where exactly one model is correct, so one model can correct the other. Oracle accuracy u is the share of problems outside the both-wrong cell, which is the accuracy a perfect selector would reach (Krogh & Vedelsby, 1994). Wrong-agreement w is the part of the both-wrong cell where the two models also return the same answer, which is the case a majority vote cannot detect. Cohen's κ (Cohen, 1960) measures how much more often than chance the two models are both correct or both wrong, so lower values mean more decoupled errors. All four numbers come from the same four cells. κ and c are therefore two readings of one measurement rather than independent evidence, and across the twelve pairs below they correlate at $r = -0.98$.

Table 4: The four outcomes for a pair of models A and B. Every problem falls into exactly one cell, and all four diversity measures are counts over these cells.

	B correct	B wrong
A correct	both correct	only A correct
A wrong	only B correct	both wrong

Three levels of decoupling. Table 5 groups pairs by what the two models differ in. *Seed only* pairs a checkpoint with itself under a different sampling seed, so the two views share every weight and differ in generation noise alone. *Same family* pairs models of one lineage across sizes or generations. *Different family* pairs models with separate architectures and pretraining data. Beyond the models of the main experiments, these pairs include Phi-3.5-mini (Abdin et al., 2024) and MiniCPM3-4B (Hu et al., 2024). The first level is what Co-RL has by construction, and the third is what a different-family cohort adds.

The three levels separate without overlap. Every different-family pair reaches $\kappa \le 0.42$ and $c \ge 29.4$, and every same-family and seed-only pair sits at $\kappa \ge 0.51$ and $c \le 24.6$. No pair falls between the two groups, at either scale. Averaged within a group, crossing families lowers κ from 0.53 to 0.38 and raises c from 23.2 to 30.8.

Table 5: Error decoupling before RL, by what the two models differ in, sorted by κ within each block.

Decoupling	Pair	$\kappa \downarrow$	$c \uparrow (\%)$	$w \downarrow (\%)$	$u \uparrow (\%)$
3B tier					
different family	Llama-3.2-3B × Phi-3.5-mini	**0.31**	**32.8**	3.0	53.0
different family	Qwen2.5-3B × Llama-3.2-3B	0.38	31.2	2.4	63.0
different family	Qwen2.5-3B × Phi-3.5-mini	0.38	31.2	4.0	55.4
different family	Qwen2.5-3B × MiniCPM3-4B	0.41	29.4	4.4	60.4
same family	Qwen2.5-3B × Qwen3-1.7B-Base	0.52	24.2	4.2	63.2
seed only	Qwen3-1.7B-Base × itself	0.52	24.0	5.0	66.4
seed only	Qwen2.5-3B × itself	0.56	22.0	4.4	62.6
7B tier					
different family	Qwen2.5-7B × Llama-3.1-8B	**0.42**	**29.4**	1.8	71.4
same family	Qwen2.5-7B × Qwen2.5-3B	0.51	24.2	3.8	69.8
same family	Qwen2.5-7B × Qwen3-1.7B-Base	0.51	24.4	4.0	70.4
seed only	Llama-3.1-8B × itself	0.51	24.6	3.0	62.0
seed only	Qwen2.5-7B × itself	0.58	19.0	5.2	74.6

Table 6: One model held fixed and the other model varied.

Peer	Decoupling	$\kappa \downarrow$	$c \uparrow (\%)$	$w \downarrow (\%)$
Anchor: Qwen2.5-3B				
itself, new seed	seed only	0.56	22.0	4.4
Qwen3-1.7B-Base	same family	0.52	24.2	4.2
MiniCPM3-4B	different family	0.41	29.4	4.4
Phi-3.5-mini	different family	0.38	31.2	4.0
Llama-3.2-3B	different family	**0.38**	**31.2**	**2.4**
Anchor: Qwen2.5-7B				
itself, new seed	seed only	0.58	19.0	5.2
Qwen2.5-3B	same family	0.51	24.2	3.8
Qwen3-1.7B-Base	same family	0.51	24.4	4.0
Llama-3.1-8B	different family	**0.42**	**29.4**	**1.8**

Holding the anchor fixed gives the same ordering. Group averages mix models of different strength, so we also fix one model and vary only the other model (Table 6). Both anchors give a monotone ladder. Relative to pairing a model with itself, a same-family model lowers κ by 0.04 and 0.07, and a different-family model lowers it by 0.18 and 0.16 while raising c by 9.2% and 10.4%. Wrong-agreement is lowest with the Llama model under both anchors, though it does not order the other rows.

D Complete Proofs

D.1 Proof of Proposition 1

In this section, we derive the reward-induced GRPO dynamics in Proposition 1. We first derive a common update expression under a fixed pseudo-label and then specialize it to self-rewarding and cross-agent supervision.

For a fixed prompt x, let $X^k = \mathbf{1}[a^k = a^\star]$ indicate whether the k-th rollout is correct. Under the binary reduction, $X^k \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(p)$, and

$$C = \sum_{k=1}^{K} X^k \sim \text{Bin}(K, p)$$

denotes the number of correct responses among the K rollouts. To analyze the induced binary dynamics, we use the log-odds

$$\ell = \log \frac{p}{1-p}$$

as a one-dimensional coordinate. Equivalently, $p = \sigma(\ell) = 1/(1 + e^{-\ell})$. We analyze the reward-induced component of GRPO in the infinitesimal-update limit, where clipping is locally inactive.

GRPO update under a fixed pseudo-label. Let $Z \in \{0, 1\}$ indicate whether a fixed pseudo-label is correct, where $Z = 1$ corresponds to $a^\star$ and $Z = 0$ to the aggregate incorrect answer. The binary reward for rollout k is

$$r^k = \mathbf{1}[X^k = Z] = (1 - Z) + (2Z - 1)X^k.$$

Since $C = \sum_{k=1}^{K} X^k$, the group-mean reward is

$$\bar{r} = (1 - Z) + (2Z - 1)\frac{C}{K},$$

and therefore

$$r^k - \bar{r} = (2Z - 1)\left(X^k - \frac{C}{K}\right). \tag{10}$$

The within-group standard deviation of the binary rewards is

$$s_K(C) \triangleq \sqrt{\frac{1}{K}\sum_{j=1}^{K}(r^j - \bar{r})^2} = \frac{\sqrt{C(K - C)}}{K}.$$

For $0 < C < K$, the normalized group-relative advantage is therefore

$$\widehat{A}^k = \frac{r^k - \bar{r}}{s_K(C)} = \frac{2Z - 1}{s_K(C)}\left(X^k - \frac{C}{K}\right).$$

For $C \in \{0, K\}$, all rewards are identical and the centered update is zero.

Treating the advantages as fixed during the policy update, the reward-induced GRPO gradient in the log-odds coordinate is

$$\begin{aligned} g_\ell(C, Z) &\triangleq \frac{\partial}{\partial \ell}\left[\frac{1}{K}\sum_{k=1}^{K}\widehat{A}^k \log \Pr_\ell(X^k)\right] \\ &= \frac{1}{K}\sum_{k=1}^{K}\widehat{A}^k \frac{\partial}{\partial \ell}\left[X^k \log p + (1 - X^k)\log(1 - p)\right] \\ &= \frac{1}{K}\sum_{k=1}^{K}\widehat{A}^k (X^k - p) \\ &= \frac{2Z - 1}{K s_K(C)}\sum_{k=1}^{K}\left(X^k - \frac{C}{K}\right)(X^k - p). \end{aligned}$$

Since $\sum_{k=1}^{K}(X^k - C/K) = 0$, the terms involving p cancel. Using $(X^k)^2 = X^k$ and $\sum_{k=1}^{K} X^k = C$ gives

$$\sum_{k=1}^{K}\left(X^k - \frac{C}{K}\right)X^k = C - \frac{C^2}{K} = \frac{C(K - C)}{K}.$$

Hence,

$$g_\ell(C, Z) = (2Z - 1)\frac{\sqrt{C(K - C)}}{K}. \tag{11}$$

Self-rewarding. Under majority-vote self-rewarding, the pseudo-label is determined by the same rollout group:

$$Z_{\text{self}} = \mathbf{1}\left[C > \frac{K}{2}\right].$$

Since K is odd,

$$2Z_{\text{self}} - 1 = \text{sign}\left(C - \frac{K}{2}\right).$$

Substituting into Eq. (11) gives

$$g_\ell(C, Z_{\text{self}}) = \text{sign}\left(C - \frac{K}{2}\right) \frac{\sqrt{C(K-C)}}{K}.$$

For a GRPO update with learning rate η,

$$\ell^+ - \ell = \eta\, g_\ell(C, Z_{\text{self}}).$$

Since $C \sim \text{Bin}(K, p)$ is random, the conditional expected one-step update is

$$\mathbb{E}[\ell^+ - \ell \mid p] = \eta\, \mathbb{E}_{C\sim\text{Bin}(K,p)}\left[\text{sign}\left(C - \frac{K}{2}\right) \frac{\sqrt{C(K-C)}}{K}\right].$$

In the infinitesimal-update limit, the corresponding mean log-odds dynamics are

$$\dot{\ell} = \eta\, \mathbb{E}_{C\sim\text{Bin}(K,p)}\left[\text{sign}\left(C - \frac{K}{2}\right) \frac{\sqrt{C(K-C)}}{K}\right].$$

Since $p = \sigma(\ell)$,

$$\begin{aligned}\dot{p} &= \frac{\partial p}{\partial \ell}\dot{\ell} \\ &= \eta\, p(1-p)\mathbb{E}_{C\sim\text{Bin}(K,p)}\left[\text{sign}\left(C - \frac{K}{2}\right) \frac{\sqrt{C(K-C)}}{K}\right],\end{aligned}$$

which recovers Eq. (2).

CO-RL: Cross-agent supervision. For agent n, let $X_n^k = \mathbf{1}[a_n^k = a^\star]$ and

$$C_n = \sum_{k=1}^{K} X_n^k \sim \text{Bin}(K, p_n).$$

Let

$$Z_{-n} = \mathbf{1}[\hat{a}_{-n}(x) = a^\star]$$

indicate whether the pseudo-label constructed from the other agents is correct. Conditional on the prompt and current policies, Z_{-n} is independent of agent n's rollout group, and hence

$$Z_{-n} \perp C_n.$$

Applying Eq. (11) to agent n gives

$$g_{\ell_n}(C_n, Z_{-n}) = \text{sign}\left(Z_{-n} - \frac{1}{2}\right) \frac{\sqrt{C_n(K-C_n)}}{K},$$

where $\ell_n = \log \frac{p_n}{1-p_n}$. Therefore,

$$\begin{aligned}\mathbb{E}[\ell_n^+ - \ell_n \mid p_n, p_{-n}] &= \eta_n\, \mathbb{E}\left[\text{sign}\left(Z_{-n} - \frac{1}{2}\right) \frac{\sqrt{C_n(K-C_n)}}{K}\right] \\ &= \eta_n\, \mathbb{E}\left[\text{sign}\left(Z_{-n} - \frac{1}{2}\right)\right] \mathbb{E}_{C_n\sim\text{Bin}(K,p_n)}\left[\frac{\sqrt{C_n(K-C_n)}}{K}\right],\end{aligned}$$

where the second equality follows from $Z_{-n} \perp C_n$.

Taking the infinitesimal-update limit and using $\partial p_n/\partial \ell_n = p_n(1-p_n)$ yields

$$\begin{aligned}\dot{p}_n &= \frac{\partial p_n}{\partial \ell_n}\dot{\ell}_n \\ &= \eta_n p_n(1-p_n)\mathbb{E}\left[\text{sign}\left(Z_{-n} - \frac{1}{2}\right)\right] \mathbb{E}_{C_n\sim\text{Bin}(K,p_n)}\left[\frac{\sqrt{C_n(K-C_n)}}{K}\right],\end{aligned}$$

which recovers Eq. (3).

Symmetric two-agent dynamics. We finally specialize the result to two symmetric agents A and B. For agent A, its pseudo-label is the majority vote of agent B's rollouts. Hence, if $C_B \sim \mathrm{Bin}(K, p_B)$,

$$Z_{-A} = \mathbf{1}\left[C_B > \frac{K}{2}\right],$$

and therefore

$$\begin{aligned}\mathbb{E}\left[\mathrm{sign}\left(Z_{-A} - \frac{1}{2}\right)\right] &= \mathbb{E}_{C_B \sim \mathrm{Bin}(K,p_B)}\left[\mathrm{sign}\left(C_B - \frac{K}{2}\right)\right] \\ &= 2V_K(p_B) - 1 = \phi_K(p_B).\end{aligned}$$

Similarly,

$$\mathbb{E}\left[\mathrm{sign}\left(Z_{-B} - \frac{1}{2}\right)\right] = \phi_K(p_A).$$

Assuming the same learning rate η for both agents and defining

$$q_K(p) = \eta p(1-p)\mathbb{E}_{C \sim \mathrm{Bin}(K,p)}\left[\frac{\sqrt{C(K-C)}}{K}\right] > 0,$$

we obtain

$$\dot{p}_A = q_K(p_A)\phi_K(p_B), \qquad \dot{p}_B = q_K(p_B)\phi_K(p_A),$$

which recovers Eq. (4).

Even K. The analysis takes K odd so that the majority vote has no ties. Our experiments use $K = 12$ for language models and $K = 8$ for vision-language models. A tie then goes to the answer sampled first, and because the rollouts are exchangeable this choice is uniform over the tied answers. For self-rewarding, a tie therefore yields the correct and the incorrect label with equal probability, and its expected contribution to Eq. (2) is zero, matching $\mathrm{sign}(0) = 0$. For cross-agent supervision, a tie only affects the accuracy of the peer label, and a majority vote over $2m$ rollouts with random tie-breaking is correct with the same probability as one over $2m - 1$ rollouts.

D.2 Proof of Proposition 2

We prove that majority-vote self-rewarding reinforces the currently favored answer and therefore induces self-confirming dynamics. For convenience, denote

$$G_K^{\mathrm{self}}(p) = \mathbb{E}_{C \sim \mathrm{Bin}(K,p)}\left[\mathrm{sign}\left(C - \frac{K}{2}\right)\frac{\sqrt{C(K-C)}}{K}\right].$$

Expanding the expectation gives

$$G_K^{\mathrm{self}}(p) = \sum_{c=0}^{K} \mathrm{sign}\left(c - \frac{K}{2}\right)\frac{\sqrt{c(K-c)}}{K}\Pr_p(C = c).$$

Since the update magnitude satisfies

$$\frac{\sqrt{c(K-c)}}{K} = \frac{\sqrt{(K-c)c}}{K},$$

each event $C = c > K/2$ can be paired with its symmetric event $C = K - c < K/2$. Moreover, the update magnitude vanishes at $c = 0$ and $c = K$. Hence,

$$G_K^{\mathrm{self}}(p) = \sum_{c=(K+1)/2}^{K-1} \frac{\sqrt{c(K-c)}}{K}\left[\Pr_p(C = c) - \Pr_p(C = K - c)\right]. \tag{12}$$

For $p \in (0,1)$ and $c > K/2$,

$$\frac{\Pr_p(C=c)}{\Pr_p(C=K-c)} = \frac{\binom{K}{c}p^c(1-p)^{K-c}}{\binom{K}{K-c}p^{K-c}(1-p)^c} = \left(\frac{p}{1-p}\right)^{2c-K},$$

where we use $\binom{K}{c} = \binom{K}{K-c}$. Since $2c-K>0$,

$$\Pr_p(C=c) - \Pr_p(C=K-c) \begin{cases} >0, & p>\frac{1}{2}, \\ =0, & p=\frac{1}{2}, \\ <0, & p<\frac{1}{2}. \end{cases}$$

The factor $\sqrt{c(K-c)}/K$ in Eq. (12) is strictly positive for $0<c<K$. Therefore, every term in the sum has the same sign, yielding

$$\operatorname{sign}\big(G_K^{\text{self}}(p)\big) = \operatorname{sign}\left(p-\frac{1}{2}\right). \tag{13}$$

We next characterize the limiting behavior. From Eq. (2),

$$\dot{p} = \eta p(1-p)G_K^{\text{self}}(p).$$

If $0<p(0)<1/2$, Eq. (13) implies $\dot{p}<0$ whenever $p \in (0,1/2)$. Thus, $p(t)$ is monotonically decreasing and bounded below by zero, and therefore converges to some $p_\infty \in [0,1/2)$.

Suppose $p_\infty > 0$. By Eq. (13),

$$p_\infty(1-p_\infty)G_K^{\text{self}}(p_\infty) < 0.$$

Since the right-hand side of Eq. (2) is continuous, $\dot{p}$ remains strictly negative in a neighborhood of p_∞, which contradicts convergence to an interior limit. Hence,

$$p(0) < \frac{1}{2} \quad \Longrightarrow \quad p(t) \to 0.$$

Similarly, if $1/2 < p(0) < 1$, then $\dot{p} > 0$. Thus, $p(t)$ is monotonically increasing and bounded above by one. The same argument rules out any interior limit, giving

$$p(0) > \frac{1}{2} \quad \Longrightarrow \quad p(t) \to 1.$$

Finally, at $p=1/2$, symmetry gives $G_K^{\text{self}}(1/2)=0$, so $p=1/2$ is an unstable equilibrium. This completes the proof.

D.3 Proof of Theorem 1

We prove Theorem 1 for the symmetric two-agent dynamics from Eq. (4),

$$\dot{p}_A = q_K(p_A)\phi_K(p_B), \qquad \dot{p}_B = q_K(p_B)\phi_K(p_A), \tag{14}$$

where

$$q_K(p) = \eta p(1-p)\mathbb{E}_{C\sim\text{Bin}(K,p)}\left[\frac{\sqrt{C(K-C)}}{K}\right] > 0$$

for $p \in (0,1)$, and $\phi_K(p) = 2V_K(p) - 1$.

Proof. **Symmetry of the dynamics.** We first establish two useful symmetries. Since the distribution of $K-C$ under $C \sim \text{Bin}(K,p)$ is $\text{Bin}(K,1-p)$ and $\sqrt{C(K-C)}$ is invariant under $C \mapsto K-C$, we have

$$q_K(1-p) = q_K(p).$$

Moreover, for odd K, complementing every rollout reverses the majority outcome, so

$$V_K(1-p) = 1 - V_K(p).$$

Therefore,

$$q_K(1-p) = q_K(p), \qquad \phi_K(1-p) = -\phi_K(p). \tag{15}$$

Since $V_K(p)$ is strictly increasing and $V_K(1/2) = 1/2$,

$$\operatorname{sign}(\phi_K(p)) = \operatorname{sign}\left(p - \frac{1}{2}\right).$$

A conserved quantity. Define

$$F_K(p) \triangleq \int_{1/2}^{p} \frac{\phi_K(u)}{q_K(u)}\, du. \tag{16}$$

Along any interior trajectory of Eq. (14),

$$\begin{aligned}\frac{d}{dt}\left[F_K(p_A) - F_K(p_B)\right] &= \frac{\phi_K(p_A)}{q_K(p_A)}\dot{p}_A - \frac{\phi_K(p_B)}{q_K(p_B)}\dot{p}_B \\ &= \phi_K(p_A)\phi_K(p_B) - \phi_K(p_B)\phi_K(p_A) \\ &= 0.\end{aligned}$$

Hence,

$$F_K(p_A) - F_K(p_B) = \text{constant} \tag{17}$$

along every trajectory.

From Eq. (15),

$$F_K(1-p) = F_K(p).$$

Furthermore, $F_K'(p) = \phi_K(p)/q_K(p)$, so F_K is strictly decreasing on $(0, 1/2)$ and strictly increasing on $(1/2, 1)$, with $F_K(1/2) = 0$.

Separatrix and basins of attraction. Consider first the disagreement region $p_A < 1/2 < p_B$. In this region,

$$\dot{p}_A > 0, \qquad \dot{p}_B < 0,$$

so the two agents move toward one another.

Suppose $p_A + p_B > 1$. Then $p_B > 1 - p_A > 1/2$. Since F_K is strictly increasing on $(1/2, 1)$ and $F_K(1-p_A) = F_K(p_A)$,

$$F_K(p_B) > F_K(1-p_A) = F_K(p_A),$$

and therefore

$$F_K(p_A) - F_K(p_B) < 0. \tag{18}$$

Because this quantity is conserved, agent B cannot reach $1/2$ before agent A: if $p_B = 1/2$, then $F_K(p_A) - F_K(p_B) = F_K(p_A) \geq 0$, contradicting Eq. (18). Thus, agent A crosses the decision boundary first.

Once $p_A, p_B > 1/2$, Eq. (14) gives

$$\dot{p}_A > 0, \qquad \dot{p}_B > 0.$$

Both probabilities therefore increase monotonically and are bounded above by one. No interior point of $(1/2, 1)^2$ is an equilibrium because $q_K(p) > 0$ and $\phi_K(p) > 0$ there. Hence,

$$(p_A(t), p_B(t)) \longrightarrow (1, 1).$$

Conversely, if $p_A + p_B < 1$, then $p_B < 1 - p_A$, and the same symmetry gives

$$F_K(p_A) - F_K(p_B) > 0.$$

The conserved quantity now prevents agent A from reaching $1/2$ before agent B. Thus, B crosses below $1/2$ first, after which both agents satisfy

$$\dot{p}_A < 0, \qquad \dot{p}_B < 0,$$

and consequently

$$(p_A(t), p_B(t)) \longrightarrow (0, 0).$$

The case $p_B < 1/2 < p_A$ follows symmetrically.

Now consider $p_A + p_B = 1$. Setting $p_B = 1 - p_A$ and using Eq. (15),

$$\begin{aligned}\dot{p}_A + \dot{p}_B &= q_K(p_A)\phi_K(1 - p_A) + q_K(1 - p_A)\phi_K(p_A) \\ &= 0.\end{aligned}$$

Hence, the line

$$p_A + p_B = 1 \tag{19}$$

is invariant. Along this line, if $p_A < 1/2 < p_B$, then $\dot{p}_A > 0$ and $\dot{p}_B < 0$; the reverse holds when $p_B < 1/2 < p_A$. Thus, every interior trajectory on this line converges to $(1/2, 1/2)$.

The cases where both agents initially lie on the same side of $1/2$ follow directly from Eq. (14). Combining all cases,

$$p_A(0) + p_B(0) > 1 \quad \Longrightarrow \quad (p_A(t), p_B(t)) \to (1, 1),$$

whereas

$$p_A(0) + p_B(0) < 1 \quad \Longrightarrow \quad (p_A(t), p_B(t)) \to (0, 0).$$

Therefore, $p_A + p_B = 1$ is the interior separatrix between the two consensus basins.

Stability of the equilibria. In a neighborhood of $(1, 1)$, both agents satisfy $p_A, p_B > 1/2$ and hence both coordinates increase monotonically toward one. Thus, $(1, 1)$ is asymptotically stable. By symmetry, $(0, 0)$ is also asymptotically stable.

Finally, consider $(1/2, 1/2)$. Since $\phi_K(1/2) = 0$, let $p_A = 1/2+\delta_A$ and $p_B = 1/2+\delta_B$. Linearizing Eq. (14) gives

$$\begin{bmatrix}\dot{\delta}_A \\ \dot{\delta}_B\end{bmatrix} = q_K(1/2)\phi'_K(1/2)\begin{bmatrix}0 & 1 \\ 1 & 0\end{bmatrix}\begin{bmatrix}\delta_A \\ \delta_B\end{bmatrix}.$$

For odd K,

$$\phi'_K(1/2) = 2K\binom{K-1}{(K-1)/2}\left(\frac{1}{2}\right)^{K-1} > 0.$$

The Jacobian therefore has eigenvalues

$$\lambda_\pm = \pm q_K(1/2)\phi'_K(1/2),$$

with corresponding eigenvectors $(1, 1)$ and $(1, -1)$. Thus, $(1/2, 1/2)$ has one unstable and one stable direction and is therefore a saddle point. Its stable manifold is exactly the invariant line $p_A + p_B = 1$ established above.

Finally, Proposition 2 gives the correct-convergence basin under independent self-rewarding as

$$\mathcal{B}^+_{\text{self}} = \left\{(p_A, p_B) : p_A > \frac{1}{2},\ p_B > \frac{1}{2}\right\}.$$

For Co-RL, the result above gives

$$\mathcal{B}^+_{\text{Co-RL}} = \{(p_A, p_B) : p_A + p_B > 1\}.$$

Therefore,

$$\mathcal{B}^+_{\text{self}} \subsetneq \mathcal{B}^+_{\text{Co-RL}},$$

which establishes the strictly larger basin of correct convergence. □

E Rephrased Training Questions

The Co-RL+ runs train one agent on the original MATH questions and the other on a rephrased copy produced by DeepSeek-R1 (DeepSeek-AI, 2025). The two copies are aligned by row and the answer is never changed. The rewrites go beyond word substitution: most of them place the problem in a concrete scenario, roughly doubling the question length. In a sample of 300 pairs, every rewrite preserved the answer and the row alignment. Two representative pairs are shown below.

Example 1 ***Answer:* 2**

ORIGINAL
How many vertical asymptotes does the graph of $y = \frac{2}{x^2+x-6}$ have?

REPHRASED
The function $f(t) = \frac{2}{t^2+t-6}$ describes the temperature of a chemical reaction over time t. How many vertical asymptotes appear on the graph of this function?

Example 2 ***Answer:* $3\sqrt{3}$**

ORIGINAL
In triangle ABC, $AB = AC = 14$ and $BC = 26$. What is the length of the shortest angle bisector in ABC? Express your answer in simplest radical form.

REPHRASED
A triangular park has two equal sides of 14 meters and a third side of 26 meters. The city plans a path from each corner that bisects its angle, and will build only the shortest one. How long is that path? Express your answer in simplest radical form.

F Complete Experiment Results

Training hyperparameters. All runs use the AdamW optimizer and sample rollouts at a temperature of 1.0. Following Co-rewarding (Zhang et al., 2026b), language models use a learning rate of 3×10^{-6}, an effective batch of 128 prompts per agent and a 3072-token cap, and train with $K = 12$ responses per prompt for 2 epochs. The Intuitor baseline on Qwen2.5-7B uses a learning rate of 1×10^{-6}. Following R1-V (Chen et al., 2025), vision-language models use a learning rate of 1×10^{-6}, a 1024-token cap and $K = 8$ responses per prompt for 1 epoch.

F.1 Three-agent Co-RL

This experiment tests whether cross-agent supervision scales to three agents and to families other than Qwen and Llama. The three runs train Qwen2.5-3B and Llama-3.2-3B-Instruct together with a third model, which is Qwen3-1.7B, Phi-4-mini-Instruct or Granite-3.3-2B-Instruct. Each agent is rewarded by the majority vote of its designated peer, as in Section 3.1. All three runs share the training and evaluation setup of the two-agent language runs. Table 7 compares each agent with the same model trained alone with GT-Reward or TTRL. The Qwen2.5-3B and Llama-3.2-3B-Instruct rows come from the run with Qwen3-1.7B, so they differ from Table 1. For Granite-3.3-2B-Instruct, the TTRL row uses its best checkpoint before the collapse. Co-RL improves all five base models and matches or exceeds GT-Reward on each.

F.2 Results on Vision-Language Models

Vision-language model families differ in both the vision encoder and the language backbone. They therefore test Co-RL under stronger heterogeneity than language models do. We train two pairs. Qwen2.5-VL-3B is paired with InternVL3.5-2B and Qwen2.5-VL-7B with InternVL3.5-8B. Each pair is trained once on open-r1 and once on MMR1 with the vision-language configuration of Section 5. We compare each model with the same model trained alone on the same data with GT-Reward or TTRL. All models are evaluated on MathVision, MathVerse, MathVista and We-Math with the protocol of Appendix F.5.

Table 8 reports the results. At 2B and 3B, Co-RL gives the best label-free average in three of the four settings and stays close to GT-Reward. It leads TTRL on at least three of the four benchmarks in every setting and on all four for Qwen2.5-VL-3B. The one setting where TTRL has the higher average is InternVL3.5-2B on MMR1. There Co-RL trails only on MathVista.

At 7B and 8B, Co-RL improves both base models on both training sets. It outperforms TTRL in three of the four settings and exceeds GT-Reward in all four. For Qwen2.5-VL-7B, it leads TTRL on every benchmark with both training sets. On open-r1 every TTRL score for Qwen2.5-VL-7B falls below the base model while every Co-RL score stays above it. Figure 10 traces this drop during training.

Table 7: Three-agent CO-RL across seven benchmarks (%).

Model	Method	GSM8K	MATH500	AMC	HEval	GPQA	MBPP	LCB	Avg
Qwen2.5-3B	Base	71.4	55.8	25.9	56.1	21.2	50.8	13.7	42.1
	GT-Reward	76.2	64.6	36.1	65.2	20.7	54.4	14.5	47.4
	TTRL	**80.4**	**66.4**	31.3	63.4	22.2	51.8	**15.9**	47.3
	CO-RL	79.8	66.3	**33.6**	**64.6**	**23.2**	**56.0**	15.8	**48.5**
Llama-3.2-3B-Instruct	Base	73.6	43.8	18.1	51.2	21.2	50.8	12.0	38.7
	GT-Reward	78.8	53.8	25.3	60.4	20.7	50.2	12.1	43.0
	TTRL	**77.9**	50.2	26.5	59.2	24.8	**51.2**	**12.0**	43.1
	CO-RL	77.8	**54.2**	**28.8**	**64.4**	**25.1**	50.9	11.7	**44.7**
Qwen3-1.7B	Base	67.0	60.9	27.5	40.0	15.3	50.6	12.4	39.1
	GT-Reward	67.1	67.0	34.3	70.1	25.2	51.2	15.2	47.2
	TTRL	**70.3**	**67.6**	32.1	**69.5**	24.8	52.0	15.1	**47.3**
	CO-RL	69.3	**67.6**	**32.7**	64.2	**27.1**	**54.6**	**15.8**	**47.3**
Phi-4-mini-Instruct	Base	80.7	68.4	33.9	72.6	19.7	51.4	14.3	48.7
	GT-Reward	84.9	74.4	36.9	73.8	21.2	54.8	13.5	51.4
	TTRL	84.6	**72.8**	36.3	**77.4**	23.7	54.6	10.0	51.4
	CO-RL	**86.4**	71.8	**38.9**	72.0	**30.3**	**55.4**	**13.7**	**52.6**
Granite-3.3-2B-Instruct	Base	63.5	54.6	24.1	32.9	25.8	45.0	10.3	36.6
	GT-Reward	73.6	56.8	23.2	45.1	20.7	46.0	10.7	39.5
	TTRL	70.4	**59.8**	**26.1**	34.8	24.2	45.6	11.0	38.8
	CO-RL	**72.9**	59.2	23.2	**42.1**	**26.8**	**46.8**	**11.2**	**40.3**

InternVL3.5-8B starts from a far stronger base and the two methods split the benchmarks more evenly. The only setting where TTRL leads on average is InternVL3.5-8B with open-r1. Together these results show that cross-agent supervision carries over to vision-language models across scales and training sets.

F.3 TRAINING DYNAMICS AND STABILITY

Text models. Figure 9 shows validation accuracy, reward standard deviation and mean completion length for Qwen2.5-3B, Llama-3.2-3B-Instruct and Llama-3.1-8B-Instruct. Figure 8(d)–(f) in the main text shows Qwen2.5-7B. Across all four backbones, CO-RL maintains a non-degenerate reward standard deviation and relatively stable completion lengths while steadily improving validation accuracy. In contrast, several self-rewarding methods become unstable during training. RENT rapidly drives the reward standard deviation toward zero and pushes completion length to the cap. Its accuracy then collapses. Intuitor similarly shows substantial reductions in reward variation and, for some models, degenerate completion lengths. TTRL keeps more reward variation, but its accuracy collapses on Qwen2.5-7B and declines steadily on Llama-3.1-8B-Instruct. On the two 3B models it stays below CO-RL for most of training. These observations match the analysis in Section 4, where supervision from a peer keeps the reward informative while self-generated supervision reinforces the agent's own errors.

Vision-language models. We observe the same qualitative behavior in the multimodal setting. As shown in Figure 10, CO-RL continues to improve evaluation accuracy while its completion length returns to the initial level, whereas TTRL eventually degrades in both accuracy and response length. More importantly, the agreement between the two agents remains well below full agreement throughout training, while the accuracy of the exchanged pseudo-labels steadily increases. The agents therefore do not converge to identical behavior. They keep meaningful differences while giving each other reliable supervision. This supports the mechanism predicted by our theory. Cross-agent learning benefits from complementary supervision and does not require the agents to collapse to the same predictions.

Table 8: Vision-language models trained separately on open-r1 and on MMR1 (%). Qwen2.5-VL-3B is trained with InternVL3.5-2B and Qwen2.5-VL-7B with InternVL3.5-8B. Base and GT-Reward are excluded from the ranking. All Co-RL rows use the Different family setting.

Backbone	Data	Method	MathVision	MathVerse	MathVista	We-Math	Avg
Qwen2.5-VL-3B	open-r1	GT-Reward	24.37	40.10	51.70	54.83	42.75
		Base	23.72	37.34	44.90	52.87	39.71
		TTRL	23.13	38.43	51.70	56.44	42.42
		Co-RL	**23.65**	**38.81**	**53.70**	**57.36**	**43.38**
	MMR1	GT-Reward	24.51	39.87	53.00	58.91	44.07
		Base	23.72	37.34	44.90	52.87	39.71
		TTRL	22.34	36.42	51.10	55.63	41.37
		Co-RL	**23.09**	**38.20**	**52.20**	**58.05**	**42.88**
InternVL3.5-2B	open-r1	GT-Reward	26.58	45.05	60.70	60.11	48.11
		Base	24.14	42.11	50.30	57.41	43.49
		TTRL	25.66	**44.44**	60.10	59.71	47.48
		Co-RL	**26.64**	43.15	**60.90**	**60.80**	**47.87**
	MMR1	GT-Reward	26.94	44.34	60.40	59.43	47.78
		Base	24.14	42.11	50.30	57.41	43.49
		TTRL	26.35	44.31	**60.50**	60.57	**47.93**
		Co-RL	**26.78**	**44.87**	58.70	**60.69**	47.76
Qwen2.5-VL-7B	open-r1	GT-Reward	27.34	47.13	69.50	65.98	52.49
		Base	26.35	46.04	69.80	65.52	51.93
		TTRL	17.01	42.34	67.10	65.34	47.95
		Co-RL	**26.74**	**49.26**	**71.30**	**70.40**	**54.42**
	MMR1	GT-Reward	28.62	48.60	67.80	68.85	53.47
		Base	26.35	46.04	69.80	65.52	51.93
		TTRL	27.01	45.18	70.70	65.23	52.03
		Co-RL	**27.70**	**48.27**	**71.30**	**66.95**	**53.56**
InternVL3.5-8B	open-r1	GT-Reward	59.70	70.81	77.60	83.74	72.96
		Base	57.66	68.78	77.10	82.18	71.43
		TTRL	60.07	**71.78**	77.20	**83.74**	**73.20**
		Co-RL	**60.23**	70.74	**78.40**	82.82	73.05
	MMR1	GT-Reward	62.43	71.80	77.50	84.14	73.97
		Base	57.66	68.78	77.10	82.18	71.43
		TTRL	**62.01**	70.94	77.80	84.14	73.72
		Co-RL	61.38	**71.52**	**78.80**	**85.11**	**74.20**

F.4 Controlling for Training and Test-Time Budgets

Co-RL trains two agents while a self-rewarding baseline trains one, so we test whether its gains come from cross-agent supervision rather than from the extra model or from ensembling. We train the same two base models independently with TTRL on the same prompts to match the training budget. At test time every prediction uses eight rollouts sampled at temperature 0.6 with top-p 0.95. A single agent votes over eight of its own rollouts, and an ensemble pools four rollouts from each of its two models. One text pair is Qwen2.5-3B with Llama-3.2-3B-Instruct. The other is Qwen2.5-7B with Llama-3.1-8B-Instruct. Both are evaluated on GSM8K, MATH-500 and AMC, whose short final answers can be compared directly by majority voting. The vision-language pair is Qwen2.5-VL-3B with InternVL3.5-2B. It is trained separately on open-r1 and on MMR1 and evaluated on the four multimodal benchmarks.

Tables 9 and 10 give the per-benchmark results, and Figure 7 plots their averages. These scores come from sampled votes, and under the matched budget Co-RL leads TTRL for every single agent and

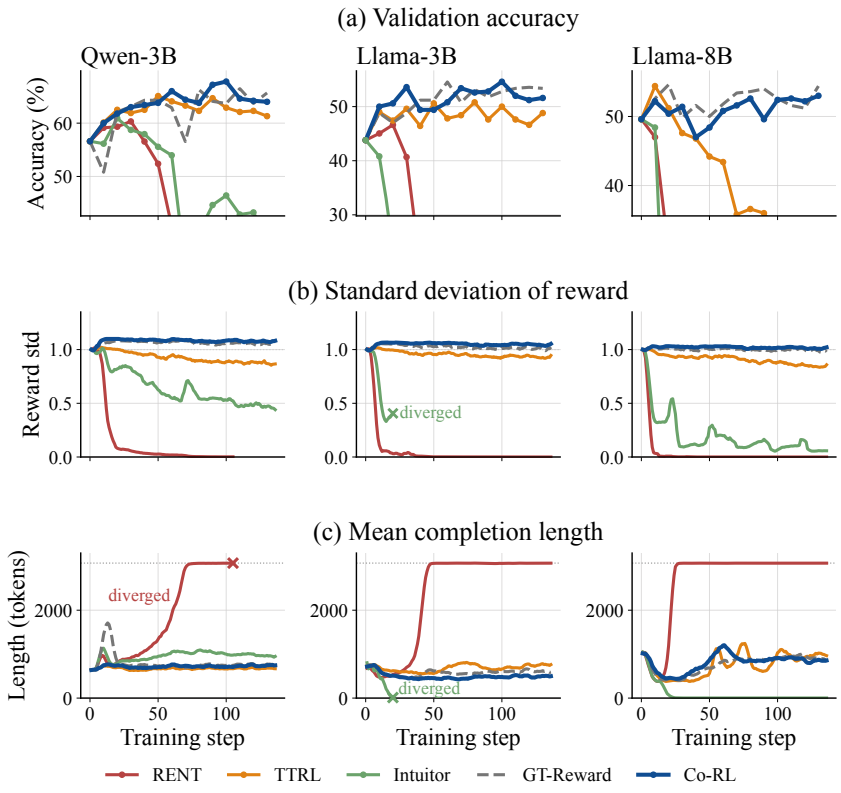


Figure 9: Training dynamics of Qwen2.5-3B, Llama-3.2-3B-Instruct and Llama-3.1-8B-Instruct, one column per backbone. (a) MATH-500 validation accuracy. (b) Reward standard deviation relative to the first step. (c) Mean completion length. The dotted line marks the 3072-token cap. Crosses mark runs that diverged and stopped early. Figure 8(d)–(f) shows Qwen2.5-7B.

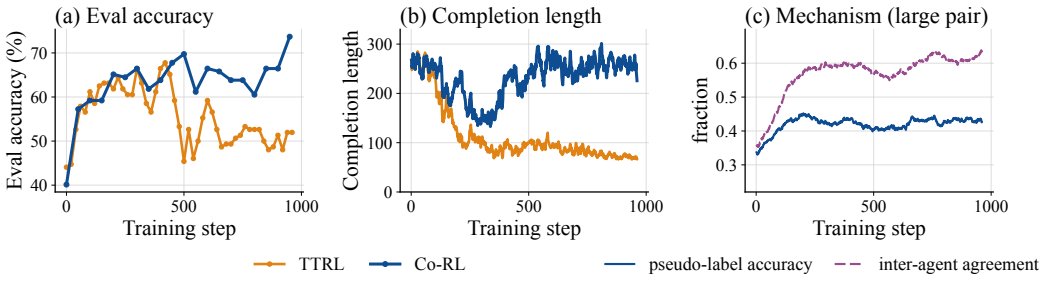


Figure 10: Training dynamics for Qwen2.5-VL-7B trained with InternVL3.5-8B on open-r1. (a) Evaluation accuracy, (b) mean completion length, and (c) the accuracy of the exchanged pseudo-labels together with the agreement between the two agents. CO-RL keeps improving and holds its completion length, while TTRL peaks and then degrades in both.

every ensemble. Its advantage thus comes neither from training two models nor from ensembling alone.

F.5 EVALUATION DETAILS

Language model evaluation. All language benchmarks except LiveCodeBench run through lm-evaluation-harness. Sampling uses a temperature of 0.6 with top-p 0.95 and a 3072-token generation budget. Trained checkpoints are evaluated with their tokenizer's chat template, which matches the

Table 9: Matched-budget comparison between TTRL and Co-RL on text reasoning benchmarks. Each block trains the same two base models with both methods. The ensemble rows pool four rollouts from each of the two models for majority voting (maj@8, $T = 0.6$). **Avg** is the macro-average over the three benchmarks. Rows are compared within a block only.

Setting	GSM8K	MATH-500	AMC	Avg
Qwen2.5-3B with Llama-3.2-3B-Instruct				
TTRL (Qwen2.5-3B)	88.2	68.8	**39.8**	65.6
TTRL (Llama-3.2-3B)	65.7	56.0	27.7	49.8
TTRL (ensemble)	88.2	68.0	38.6	64.9
Co-RL (Qwen2.5-3B)	87.4	**72.8**	37.4	65.9
Co-RL (Llama-3.2-3B)	87.3	58.8	33.7	59.9
Co-RL (ensemble)	**90.1**	70.8	**39.8**	**66.9**
Qwen2.5-7B with Llama-3.1-8B-Instruct				
TTRL (Qwen2.5-7B)	88.3	**80.2**	44.6	71.0
TTRL (Llama-3.1-8B)	87.0	59.6	34.9	60.5
TTRL (ensemble)	89.5	76.4	43.4	69.7
Co-RL (Qwen2.5-7B)	**90.8**	80.0	48.2	**73.0**
Co-RL (Llama-3.1-8B)	87.8	62.0	41.0	63.6
Co-RL (ensemble)	90.6	76.8	**49.4**	72.3

Table 10: Matched-budget comparison between TTRL and Co-RL on multimodal reasoning benchmarks. Both settings train the same two base models, Qwen2.5-VL-3B and InternVL3.5-2B. The ensemble rows pool four rollouts from each of the two models for majority voting (maj@8, $T = 0.6$, top-p 0.95). Rows are grouped by training set.

Setting	MathVision	MathVerse	MathVista	We-Math	Avg
open-r1					
TTRL (Qwen2.5-VL)	22.96	31.45	61.10	63.39	44.73
TTRL (InternVL3.5)	29.67	**38.91**	62.30	67.24	49.53
TTRL (ensemble)	27.24	35.13	65.40	67.41	48.80
Co-RL (Qwen2.5-VL)	25.43	35.66	64.80	65.80	47.92
Co-RL (InternVL3.5)	**30.46**	38.60	63.30	67.87	50.06
Co-RL (ensemble)	28.95	38.48	**67.00**	**69.08**	**50.88**
MMR1					
TTRL (Qwen2.5-VL)	17.27	30.71	63.40	60.57	42.99
TTRL (InternVL3.5)	28.78	39.47	63.70	66.90	49.71
TTRL (ensemble)	25.53	37.77	67.00	66.44	49.19
Co-RL (Qwen2.5-VL)	25.86	34.59	66.00	64.94	47.85
Co-RL (InternVL3.5)	**30.79**	**40.94**	65.30	67.53	51.14
Co-RL (ensemble)	30.49	39.75	**69.40**	**69.54**	**52.30**

prompt format used during training. For AMC we sample eight responses per problem and report avg@8, averaged over three evaluation seeds. All other benchmarks use a single sample. GPQA uses the Diamond subset with a boxed-answer prompt. Answers are scored by rule-based graders with math-aware normalization, and multiple-choice questions accept both the option letter and the option value. LiveCodeBench (release v6) runs through its official harness at its default temperature of 0.2 with one sample per problem. The harness hard-codes a gated Llama-3 tokenizer for prompt construction, and we substitute the openly available Llama-3.2-3B-Instruct tokenizer, which carries the same chat template.

CoMAS comparison. The comparison with CoMAS uses its benchmark suite (GSM8K, MATH-500, HumanEval, MBPP, SciBench (Wang et al., 2024b), GPQA and MMLU (Hendrycks et al., 2021a)),

its driver and its graders, with training prompts drawn from their blended 2,000 problems from MATH (Hendrycks et al., 2021b), KodCode (Xu et al., 2025b) and WebInstruct-verified (Ma et al., 2025). Their protocol draws five samples per question at temperature 0.7 and then issues a sixth call that reasons over the five drafts, and only the sixth response is graded.

Vision-language evaluation. We follow the benchmark splits of MM-UPT, the MathVision test set (3,040 problems), the MathVerse testmini split (3,940, all five versions), the MathVista testmini split (1,000) and the We-Math testmini split (1,740). Decoding is greedy with a 16384-token generation budget. Images are resized so that the long side does not exceed 1024 pixels, matching the training-time preprocessing. Trained checkpoints are prompted in the format they were trained on, with reasoning in think tags and the final answer in answer tags. Base models receive a standard boxed-answer prompt. Scoring is two-stage. A rule-based pass extracts the final answer and grades it with math-aware matching, and a response that never commits to an answer in a recognized format counts as incorrect. Responses that follow the format but fail the rule match are passed to an LLM judge, Qwen2.5-32B-Instruct at temperature 0, which accepts only semantically equivalent answers and rejects responses that are cut off.

MM-UPT comparison. Table 3 follows the evaluation protocol of MM-UPT so that the rows reported from their paper are comparable to ours. Responses are sampled at temperature 0.2 with top-p 1.0, and answers are extracted and graded by Qwen2.5-32B-Instruct as a judge on the benchmark splits used in the MM-UPT paper.

Engineering notes. All fixes below are included in our code. None of them changes the training or evaluation semantics. They repair crashes or a wrong backend choice in the underlying libraries.

Qwen2.5-VL, vision-tower attention backend. In vLLM 0.11.2, the helper that selects the vision tower's attention backend silently promotes xFormers to the bundled FlashAttention build. That build supports head dimensions that are multiples of 32 only, and the Qwen2.5-VL vision tower has head dimension 80, so the model crashes at load. We patch the helper to keep the original xFormers choice, which has no head-dimension restriction. The InternVL vision tower has a head dimension that is a multiple of 32 and is unaffected, and later vLLM releases fix the bug.

InternVL3.5, processor and tiling. We use the transformers-native HF variants, whose checkpoints load through `AutoProcessor` without the legacy remote-code path. Dynamic patch tiling is disabled at both training and evaluation, so the image-token count per sample is identical in the two settings. Model families are detected from each checkpoint's configuration file rather than from directory names, since CO-RL run directories contain the names of both models.