

One Frozen Simulator Is Not Enough: Simulator Collapse in Multi-Agent RL

Simon Yu¹ Nicholas Tomlin² Marwa Abdulhai³ Ximing Lu⁴ Derek Chong⁵
Abe Hou⁵ Dilara Soylu⁵ Sergey Levine³ Christopher D. Manning⁵ Weiyan Shi¹

¹Northeastern University ²New York University ³UC Berkeley

⁴University of Washington ⁵Stanford University

Abstract

Multi-agent reinforcement learning (RL) for human-AI interaction typically trains the agent against a single large language model (LLM) that simulates the user. We show that this approach systematically fails to generalize, and we trace the failure to **simulator collapse**. Because the simulator LLM is mode-collapsed, a policy trained against it overfits to narrow strategies that exploit the simulator’s dominant mode. Such a policy transfers poorly to unseen simulators and real users. We formalize this collapse and propose two complementary solutions, one at inference time and one at training time. Verbalized Sampling reduces the frozen simulator’s mode collapse by drawing each reply from a verbalized response distribution. Co-Training updates the user simulator on the same rollouts as the policy, so the simulator’s mode moves as the policy improves. A population variant picks the simulator for each rollout from a pool of recent checkpoints. We validate both solutions on three multi-turn benchmarks: Persuasion for Good, τ^2 -bench, and CooperBench. Verbalized Sampling improves held-out success by up to 9% over single-simulator RL, and Co-Training pushes gains further to 14%; the human study shows similar gain on real users. Both solutions preserve the policy diversity that collapses under single-simulator RL. We release SCOPE, an open-source framework for Population Co-Training in multi-agent RL.¹

1 Introduction

Reinforcement learning with verifiable rewards has improved LLMs on single-turn tasks such as mathematical reasoning (DeepSeek-AI, 2025; Liu et al., 2025d; Yu et al., 2025b), tool use (Feng et al., 2025; Jin et al., 2025), and software engineering (Yang et al., 2025b; Wei et al., 2025). In each, the environment verifies the policy’s answer before the update.

¹Code: https://github.com/CHATS-lab/scope_usim

Work on multi-turn human-AI interaction (Qian et al., 2025; Zhou et al., 2025a; Wu et al., 2026) covers customer support (Barres et al., 2025), collaborative coding (Khatua et al., 2026; Wang et al., 2026b), persuasion (Wang et al., 2019), and tutoring (Abdulhai et al., 2025). Training against real users at the scale RL requires is prohibitively slow and expensive, so prior work has turned to LLM-based user simulators (Park et al., 2023, 2024).

A common setup prompts a single frozen LLM to play the user (Qian et al., 2025; Yu et al., 2025a). In multi-turn RL (Abdulhai et al., 2023), this simulated user takes the place of the deterministic verifier. Its output distribution decides which states the policy visits and which gradients the policy receives. Policies trained this way can systematically fail to generalize. We call the underlying failure mode **simulator collapse**. Many aligned LLM simulators are mode-collapsed (Jiang et al., 2025; GX-Chen et al., 2025; Zhang et al., 2025a). A policy trained against one of them receives gradients dominated by that simulator’s modal behavior and overfits to narrow strategies that exploit the mode (MacDiarmid et al., 2025).

Our analysis in Section 3 shows that a mode-collapsed simulator biases the policy gradient toward its mode even when the gradient does not vanish. Repeated updates rank the policy’s trajectories by how well they exploit that mode and concentrate probability on a narrow exploit set. This explains the rapid policy-entropy drop we observe in training. The resulting policy does well in distribution but fails on unseen simulators whose responses include behaviors the training simulator did not produce.

We propose two solutions (Section 3.4). **Verbalized Sampling** (Zhang et al., 2025a) restores within-simulator diversity without retraining. At each simulator turn of a rollout, the frozen simulator is asked for a verbalized response distribution from which one response is sampled. **Co-Training**

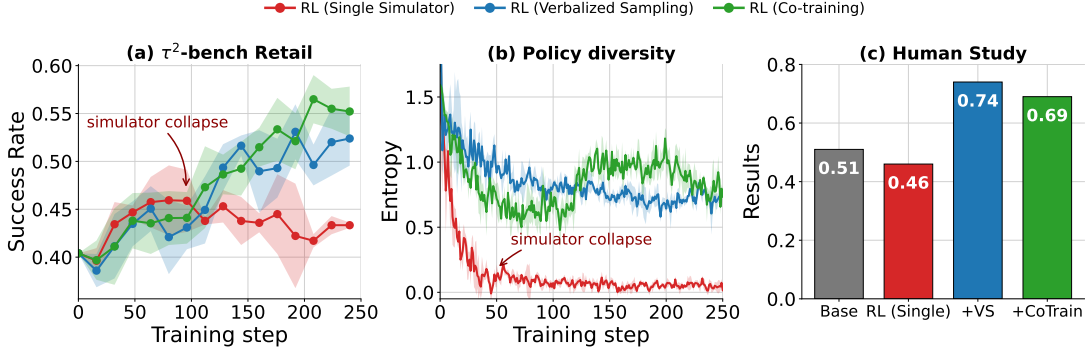


Figure 1: **Single-simulator RL collapses; Verbalized Sampling and Co-Training recover.** τ^2 -bench (Qwen3-4B-Instruct). (a) Held-out generalization: RL against a single frozen simulator peaks early, then starts collapsing. (b) Policy entropy: the same recipe drives the policy’s entropy to near zero. (c) Human study: both fixes lift real-user performance over RL (Single), which drops below even the untrained baseline.

updates the user simulator alongside the policy on the same conversations, and the two models co-evolve (Liu et al., 2025c,a). As the policy improves, the simulator’s mode at each history shifts. A strategy that exploited an earlier mode then stops winning against the current one. **Population Co-Training** samples the simulator for each rollout from a pool of recent checkpoints. On CooperBench the trained partner is a second coding agent.

Section 4 validates both solutions across three settings: Persuasion for Good (Wang et al., 2019) (adversarial dialogue), τ^2 -bench (Barres et al., 2025) (collaborative tool-calling), and CooperBench (Khatua et al., 2026) (collaborative software engineering). Across all three benchmarks, Population Co-Training reaches the highest held-out task success and preserves policy entropy.

Our contributions are:

1. **Identifying simulator collapse.** We formalize how a single frozen, mode-collapsed user simulator biases the policy gradient toward its mode. Under this bias the policy concentrates on a narrow, simulator-specific exploit set, and its entropy collapses (§3.2).
2. **Two solutions at different points in the training loop.** Verbalized Sampling restores the per-turn diversity of a frozen simulator. Co-Training jointly updates the policy and a trainable user simulator on the same rollouts, so the partner adapts as the policy improves (§3.4). We release SCOPE, an open-source framework that supports multi-model rotation, self-play, and dual-model Co-Training through one pluggable interface.
3. **Empirical validation across three multi-agent RL settings.** On Persuasion for Good, τ^2 -bench,

and CooperBench, single-simulator RL drops back toward the untrained baseline. Both Verbalized Sampling and Co-Training close most of the held-out gap, and Population Co-Training yields the strongest held-out task success rate (§4). We additionally run a human study on τ^2 -bench and Persuasion for Good, where Co-Training improves task outcome and both methods improve P4G dialogue naturalness over single-simulator RL (Appendix D).

2 Background

Multi-agent RL as a POMDP. We model multi-turn dialogue as a two-player partially observable Markov decision process (POMDP) whose state is the shared conversation history. Persuasion for Good and τ^2 -bench are partially observable stochastic games (POSGs), in which the user simulator has private observations (goal, persona) and only the agent receives task reward. In CooperBench, a decentralized POMDP (Dec-POMDP), two cooperating coding agents share a task-success reward. One shared-history POMDP covers all three. The theory in §3.2 needs only the joint trajectory distribution, which is invariant to how private observations are partitioned. At turn t the state is $s_t = (o_0, a_0^\pi, a_0^\phi, \dots, o_{t-1}, a_{t-1}^\pi, a_{t-1}^\phi)$. After the agent samples an utterance $a_t^\pi \sim \pi_\theta(\cdot | s_t)$, the user simulator samples a response $a_t^\phi \sim \phi_\psi(\cdot | s_t, a_t^\pi)$. A trajectory $\tau = (s_0, a_0^\pi, a_0^\phi, \dots, s_T)$ has terminal reward $R(\tau)$, and the agent maximizes $J(\theta; \psi) = \mathbb{E}_{\tau \sim (\pi_\theta, \phi_\psi)}[R(\tau)]$. The state-visitation distribution $d_\psi^{\pi_\theta}(s) = \sum_t \Pr(s_t = s | \pi_\theta, \phi_\psi)$ depends on both the agent and the simulator. The simulator therefore affects the reward of each trajec-

tory and also which histories the policy learns from. When the simulator is fixed we write $J_\phi(\theta) = J(\theta; \psi)$.

Policy update. We apply REINFORCE (Williams, 1992) to full multi-turn trajectories with group-relative reward normalization. For each task we sample G trajectories, score each by its terminal reward $R(\tau^n)$, and assign the z-scored advantage $\hat{A}^n = (R(\tau^n) - \bar{R})/\sigma_R$ to every agent token in trajectory n . If all trajectories in a group receive the same reward, $\sigma_R = 0$ and the group contributes no update. When $\sigma_R > 0$ but the simulator keeps responding the same way at the histories visited during training, the remaining contrast mostly ranks agent samples by how well they exploit that simulator. The gradient is then nonzero but biased (Section 3).

3 Simulator Collapse

Aligned LLMs favor typical responses under direct prompting (Jiang et al., 2025; Zhang et al., 2025a,b; GX-Chen et al., 2025). User-simulation studies find LLM simulators overly cooperative and stylistically uniform (Naous et al., 2025; Zhou et al., 2026; Mehri et al., 2026).

Collapse on the policy’s training rollouts brings the policy gradient close to the gradient against a deterministic mode-user simulator (§3.2). Group-relative updates then move policy mass onto the narrow strategy that wins against the mode, and policy entropy drops. §3.3 tests both predictions. Co-Training (§3.4) counters collapse by training the simulator so that its mode shifts.

3.1 Definition and Hypothesis

Mode collapse. At a simulator turn, the user response is sampled from $\phi_\psi(\cdot \mid s_t, a_t^\pi)$. We call the most likely response at that turn the simulator’s *mode*:

$$a_\phi^*(s, a^\pi) \in \arg \max_{a^\phi} \phi_\psi(a^\phi \mid s, a^\pi), \quad (1)$$

and write $\epsilon_\phi(s, a^\pi) = 1 - \phi_\psi(a_\phi^*(s, a^\pi) \mid s, a^\pi)$ for the probability that the simulator deviates from its mode. Aligned LLMs concentrate strongly on the mode, an effect often called mode collapse (Jiang et al., 2025; Zhang et al., 2025a,b; GX-Chen et al., 2025). Small ϵ_ϕ means the simulator keeps emitting its mode.

Definition 3.1 (Simulator collapse on the training rollouts). For a policy π_θ and a threshold $\epsilon^* \in$

$[0, 1]$, we say the simulator is ϵ^* -collapsed on the rollouts if $\mathbb{E}_{(s_t, a_t^\pi) \sim (\pi_\theta, \phi_\psi)} [\epsilon_\phi(s_t, a_t^\pi)] \leq \epsilon^*$ at the simulator turns visited by rollouts from (π_θ, ϕ_ψ) .

The definition refers only to the histories the current policy visits during training. We state ϵ_ϕ over exact token sequences. All results also hold after a measurable coarsening onto behavior classes (Remark A.1). The threshold ϵ^* enters Theorem 3.2, whose bound is tight when $\bar{\epsilon}_H(\theta) \leq \epsilon^* H$. As an empirical proxy we track the fraction of zero-variance batches, which rises above 85% during single-simulator training (Figure 19). The proxy is one-sided. Small ϵ_ϕ implies small reward variance (Lemma A.3), but sparse rewards also produce zero variance.

Hypothesis. A mode-collapsed simulator reduces multi-agent RL to training against one fixed user. The RL policy learns the strategy that wins against the simulator’s mode. When other models or real users deviate from the mode, the strategy fails.

3.2 How Simulator Collapse Affects the Policy

Mode collapse biases the policy gradient toward a deterministic mode-user objective (Theorem 3.2). It also removes simulator-side reward variance. Without that variance, the z-scored advantage ranks samples by how well they exploit the mode and ignores how they fare against other user behaviors (Lemma A.3). Policy mass then concentrates geometrically on the mode-exploit set A_x (Proposition 3.3, Corollary 3.4). The resulting low-entropy policy underperforms on real users whose behaviors fall outside A_x (Proposition A.9). Proofs are in Appendix A.

Gradient bias under collapse. Let M_ϕ be the dialogue POMDP induced by ϕ , and let M_{mode} be the modified POMDP in which every simulator turn deterministically emits the mode $a_\phi^*(s_t, a_t^\pi)$, with objective $J_{\text{mode}}(\theta)$. M_{mode} depends only on ϕ_ψ and does not change with π_θ .

Theorem 3.2 (Simulator collapse induces mode-user optimization). *Assume rewards are bounded in $[0, R_{\max}]$ and the trajectory-level policy score satisfies $\|\sum_t \nabla_\theta \log \pi_\theta(a_t^\pi \mid s_t)\| \leq B$ (e.g., under finite-length truncation and gradient clipping; see Appendix A.2). Couple π_θ in M_ϕ and M_{mode} turn by turn (same task, same agent randomness, maximal coupling at each simulator turn). Define the accumulated collapse error along the rollout*

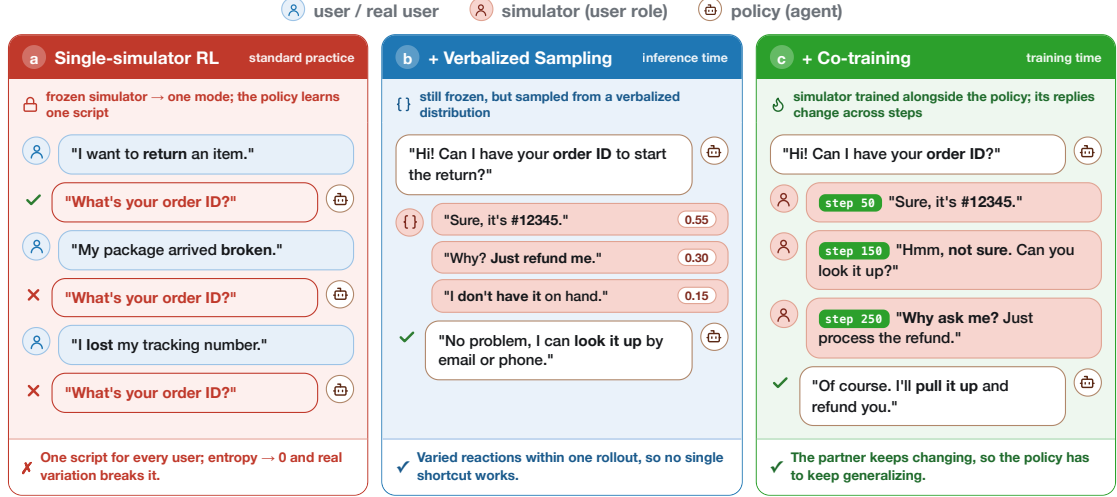


Figure 2: **Simulator collapse and the two solutions.** (a) **Problem.** A frozen LLM simulator covers one mode of a broad real-user distribution. The RL policy fits that mode and gives the same narrow reply to real users outside it. (b) **Verbalized Sampling (inference time).** One prompt asks the frozen simulator for several plausible user replies with likelihoods, so a single rollout contains varied reactions (acceptance, pushback, refusal). (c) **Co-Training (training time).** The simulator is trained during RL, and the mode the policy would fit keeps moving.

as

$$\bar{\epsilon}_H(\theta) = \mathbb{E} \left[\sum_{t=1}^H \epsilon_\phi(s_t, a_t^\pi) \right]. \quad (2)$$

Write P_ϕ^θ and P_{mode}^θ for the joint trajectory distributions under π_θ in M_ϕ and M_{mode} respectively. Then $D_{\text{TV}}(P_\phi^\theta, P_{\text{mode}}^\theta) \leq \bar{\epsilon}_H(\theta)$, and

$$\|\nabla_\theta J_\phi(\theta) - \nabla_\theta J_{\text{mode}}(\theta)\| \leq 2BR_{\text{max}} \bar{\epsilon}_H(\theta). \quad (3)$$

By Theorem 3.2, the gradient stays nonzero but is biased, up to $\bar{\epsilon}_H(\theta)$, toward the objective in which every user emits the mode a_ϕ^* . The bound covers the idealized gradient under the bounded trajectory-score assumption. Our implemented update adds group-relative normalization and gradient clipping (Appendix B.1). For that update, the theorem indicates the direction of the bias but does not tightly bound the actual surrogate loss. The bound is informative when $\bar{\epsilon}_H \ll 1$ (Appendix A.4).

Policy concentration under a persistent mode advantage. Let Y be an agent-strategy abstraction that coarsens token sequences into a small strategy set (e.g., “open with empathy,” “cite statistics”), with distribution $q_k(y | x)$ at update k . A strategy y exploits the simulator’s mode if its mode-user value $Q_{\text{mode}}(x, y)$ is high. A_x denotes the mode-exploit set and $\Delta_x > 0$ its gap over non-exploit strategies (Appendix A.6). We model each policy update as a stylized KL-regularized softmax step. This step

is a sufficient mechanism for the entropy collapse we observe, and the resulting proposition does not apply directly to the implemented optimizer.

Proposition 3.3 (Mode advantage concentrates policy mass). *Under a KL-regularized softmax policy-gradient step on $q_k(y | x)$ with learning rate η and bounded Q -estimation errors, the log-odds of A_x satisfy*

$$\log \frac{q_{k+1}(A_x | x)}{q_{k+1}(A_x^c | x)} \geq \log \frac{q_k(A_x | x)}{q_k(A_x^c | x)} + g_x, \quad (4)$$

where $g_x > 0$ depends on η , the gap Δ_x , and the simulator-collapse and estimation errors (Appendix A.6 gives exact constants).

Corollary 3.4 (Entropy concentration). *Iterating Eq. 4 gives*

$$q_k(A_x | x) \geq \frac{1}{1 + \frac{1 - q_0(A_x | x)}{q_0(A_x | x)} e^{-kg_x}}, \quad (5)$$

so the strategy distribution concentrates onto A_x geometrically fast in k .

Strategy concentration shows up as a drop in the token-level entropy we measure in §3.3 only to the extent that the exploit strategies in A_x are themselves low-entropy text (fixed scripts, formulaic appeals). We see this regime qualitatively in late-training batches, where the three transcripts sampled from one context become nearly word-for-word the same (Appendix E.9).

Transfer to real users. Proposition A.9 (Appendix A.8) lower-bounds the deployment regret of the concentrated policy on real users whose behaviors fall outside A_x , as a function of the real-user mass on these missing behaviors and the mode-exploit gap. §3.3 tests this regime empirically.

3.3 Empirical Evidence

We examine these results empirically on Persuasion for Good (Wang et al., 2019) and τ^2 -bench (Barres et al., 2025). For each single-simulator run we track training reward, policy entropy, and eval reward on a held-out panel Φ_{eval} of six simulators spanning seen and unseen families. Setup details are in Appendix B.4.

Single-simulator RL exhibits simulator collapse in practice. We train against three frozen simulators of varying modal concentration (GPT-5-mini, Haiku-4.5, Gemini-3-Flash). Figure 3 shows the training results. Training reward climbs in every run, fastest for the most modal simulator. OOD eval peaks early and turns over, but the magnitudes differ sharply: the most modal simulator (Gemini-3-Flash) crashes below the untrained baseline, while the least modal (GPT-5-mini, the one we use in our main experiments) sees a gentler but still clear decline. Policy entropy crashes toward zero across all three runs. The failure is on the policy side: the policy learns the narrow exploit and fails to transfer to unseen simulators. The token-entropy pattern is consistent with the strategy-level concentration that Corollary 3.4 predicts, under the condition that the exploit strategies are themselves low-entropy (Remark A.5).

Takeaway. The failure comes from the training environment, instead of the algorithm. Two solutions follow, at different points in the training loop. **Verbalized Sampling** (Zhang et al., 2025a) is the inference-time solution: we draw from a verbalized response distribution at each turn during rollout. **Co-Training** is the training-time solution: we update the simulator alongside the policy, so the mode the policy could lock onto shifts as training proceeds. The next section develops both solutions; their empirical comparison is in §4.2.

3.4 Verbalized Sampling and Co-Training

The analysis in §3.2 rests on two assumptions. First, the per-turn collapse error $\epsilon_\phi(s_t, a_t^\pi)$ stays near zero. In that case $\bar{\epsilon}_H(\theta)$ in Theorem 3.2 approaches zero and $\nabla_\theta J_\phi$ approaches $\nabla_\theta J_{\text{mode}}$. Second, the

mode-exploit set A_x stays fixed during training, which lets the log-odds of A_x accumulate over k updates until Corollary 3.4 concentrates q_k on A_x at geometric rate g_x . Each solution removes one assumption. Under the reference-recovery assumption ($D_{\text{TV}}(p_\phi^{\text{VS}}, P) \leq \eta$, Proposition 3.5), Verbalized Sampling brings the policy gradient close to the reference-user gradient, away from the mode-user gradient that Corollary 3.4 requires. Co-Training moves A_x at every step. The log-odds then have no fixed target to accumulate toward.

Verbalized Sampling. A greedy query drives $\epsilon_\phi(s)$ toward zero. A verbalized query returns K candidate responses with verbalized probabilities, and their distribution $p_\phi^{\text{VS}}(\cdot | s)$ approximates the simulator’s pre-RLHF reference distribution $P(\cdot | s)$ (Zhang et al., 2025a). The distribution-level prompt elicits the response distribution that direct aligned prompting sharpens. The closeness $D_{\text{TV}}(p_\phi^{\text{VS}}, P) \leq \eta$ is an empirical assumption. Appendix E.5 discusses when it holds and when it fails. The reference P comes from the simulator’s pretraining and differs from the real-user population P_{real} . The proposition below makes no claim about distance to real users, which the human study (Appendix D) tests separately.

Proposition 3.5 (Reference-gradient recovery under Verbalized Sampling). *If $D_{\text{TV}}(p_\phi^{\text{VS}}(\cdot | s, a^\pi), P(\cdot | s, a^\pi)) \leq \eta(s, a^\pi)$ at every visited state along an H -turn rollout, then the trajectory distributions and policy gradients in M_{VS} vs. the reference-user environment M_{ref} satisfy $D_{\text{TV}}(P_{\text{VS}}^\theta, P_{\text{ref}}^\theta) \leq \bar{\eta}_H(\theta)$ and*

$$\|\nabla_\theta J_{\text{VS}}(\theta) - \nabla_\theta J_{\text{ref}}(\theta)\| \leq 2BR_{\text{max}} \bar{\eta}_H(\theta),$$

where $\bar{\eta}_H(\theta) = \mathbb{E}[\sum_{t=1}^H \eta(s_t, a_t^\pi)]$.

Proposition 3.5 has the same form as Theorem 3.2, with the reference-user gradient in place of the mode-user gradient. If P has non-trivial mass away from its mode and η is small, VS moves training out of the mode-oracle regime that Corollary 3.4 requires. Appendix E.5 proves the bound and discusses the limits of the reference-recovery assumption. Its γ -sharpening result shows that direct aligned prompting suppresses tail behaviors exponentially, whereas VS preserves them.

Co-Training. We update the user simulator on its own turns of each conversation, so one rollout trains both sides. As its mode at each history

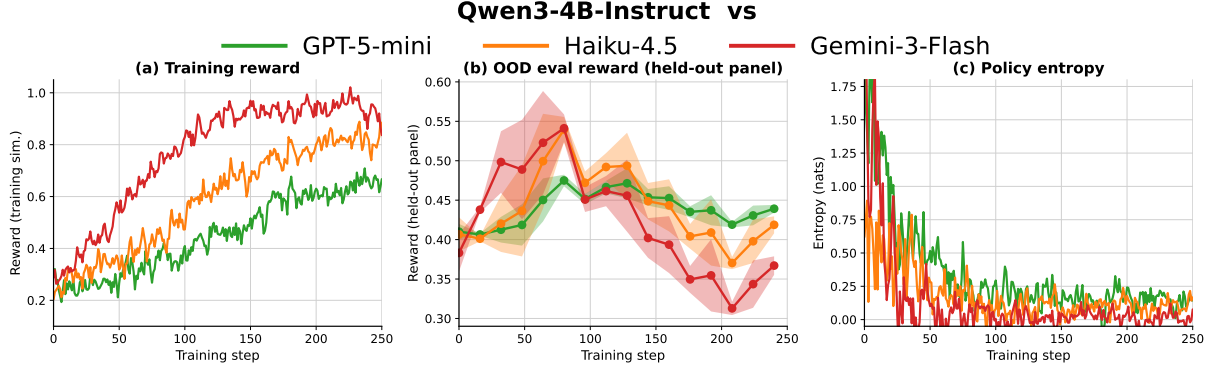


Figure 3: **Single-simulator RL exhibits simulator collapse.** Three single-simulator REINFORCE runs, three seeds each, $\pm 1\sigma$ shading on OOD. Training reward climbs in every run (a), OOD eval peaks early and declines (b), and policy entropy collapses (c). The decoupling between (a) and (b) is simulator collapse passing through into policy collapse; (c) is the mechanism.

shifts during training, a strategy tuned to an earlier mode stops winning. With A_x no longer constant, the geometric concentration of Corollary 3.4 does not apply. Appendix A.7 formalizes this with exclusive-lead counters. The simulator’s reward must keep it in the informative-variation regime (Remark A.12) and prevent collapse onto a different mode. §4.1 gives the per-task rewards, and Appendix E.8 shows that both reward extremes underperform. For binary rewards the curriculum targets a success rate of ≈ 0.5 , where the within-batch variance $p(1-p)$ peaks at $\sigma^2 = 0.25$ and group-relative advantages have the widest spread.

4 Experiments

The theory motivates four questions. **(Q1)** Does single-simulator RL collapse? **(Q2)** Do Verbalized Sampling and Co-Training recover the signal at the complementary layers they target? **(Q3)** Does co-evolution also break the fixed-partner ceiling in symmetric cooperation? **(Q4)** Do the LLM-panel gains transfer to real users?

4.1 Setup

Tasks. Three multi-turn benchmarks: **Persuasion for Good (P4G)** (Wang et al., 2019), a persuader arguing for a charitable donation against a resistant donor (reward $r = \min(\text{donation}/2, 1)$); τ^2 -**bench** (Barres et al., 2025), customer-service dialogues on retail or airline tasks with binary per-split success; and **CooperBench** (Khatua et al., 2026), two coding agents coordinating on a multi-step task with binary success. CooperBench uses Qwen3.5-9B and Qwen3.5-27B since smaller models cannot complete the tasks. P4G uses its

donation-based adversarial reward and CooperBench its symmetric task-success reward; on τ^2 -bench the simulator is trained with a SPICE-style *curriculum* reward (Liu et al., 2025b) that targets within-batch variance $\sigma^2 \approx 0.25$; this choice is essential, since adversarial and cooperative simulator rewards both collapse the simulator onto a new mode and drop eval reward (Appendix E.8). We evaluate every 16 steps and pick the best-mean-panel-score checkpoint. P4G and τ^2 -bench use a 6-simulator panel (3 seen during training plus 3 unseen families); CooperBench uses symmetric cross-play against Claude Haiku 4.5, GPT-5, and Gemini-3-Flash.

Training. All methods share the policy update from §2. For Co-Training, both sides update from the same rollout, the simulator on a task-appropriate objective. The population variant additionally samples the active simulator from a pool of recent checkpoints (full buffer specification in Appendix B.1). The framework SCOPE implements all paradigms. Methods differ in per-step training compute: frozen-simulator methods (VS, Ensemble, Persona-Guided) update only the agent, while Co-Training and Population Co-Training update both sides on the same rollout.

Baselines. Six paradigms under matched setup: *Base* (untrained); *RL (Single)* against GPT-5-mini, the standard recipe; *Ensemble Models*, cycling $K=3$ frozen models from different families. Plus a *Persona-Guided* simulator, which conditions a single GPT-5-mini simulator on personas (Abdulai et al., 2026). **Verbalized Sampling** (Zhang et al., 2025a), sampling from the simulator’s ver-

Table 1: **User-simulator setting (P4G, τ^2 -bench)**. Best held-out-panel score across training; subscript = panel-std over six held-out simulators. P4G reward $r = \min(\text{donation}/2, 1)$; τ^2 -bench reports per-split success rate (%). **Bold** = best per model size; underline = second best. “N/A” on the P4G column for Persona-Guided: the P4G task already conditions on personas, so the baseline does not apply.

Model	Method	P4G	τ^2 -bench	
		Reward $\uparrow$	Retail $\uparrow$	Airline $\uparrow$
Qwen3-4B-Instruct	Base	0.216 $\pm$ 0.03	40.4 $\pm$ 3.1	24.0 $\pm$ 2.8
	RL (Single)	0.275 $\pm$ 0.12	46.1 $\pm$ 5.2	29.8 $\pm$ 4.3
	+ Persona-Guided	N/A	49.2 $\pm$ 3.2	31.6 $\pm$ 3.0
	+ Ensemble Models ($K=3$)	0.394 $\pm$ 0.04	57.1 $\pm$ 3.4	40.1 $\pm$ 3.6
	+ Verbalized Sampling	0.484 $\pm$ 0.05	55.5 $\pm$ 3.8	36.9 $\pm$ 3.5
	+ Co-Training	0.438 $\pm$ 0.05	<u>60.5</u> $\pm$ 3.9	44.4 $\pm$ 4.0
	+ Population Co-Training	0.508 $\pm$ 0.04	62.2 $\pm$ 3.6	45.7 $\pm$ 3.7
Qwen3-8B	Base	0.253 $\pm$ 0.03	48.1 $\pm$ 3.0	30.2 $\pm$ 2.9
	RL (Single)	0.342 $\pm$ 0.11	52.5 $\pm$ 6.1	35.2 $\pm$ 5.9
	+ Persona-Guided	N/A	55.7 $\pm$ 3.1	37.4 $\pm$ 3.0
	+ Ensemble Models ($K=3$)	0.450 $\pm$ 0.04	62.4 $\pm$ 3.3	43.6 $\pm$ 3.5
	+ Verbalized Sampling	0.587 $\pm$ 0.05	60.7 $\pm$ 3.5	40.2 $\pm$ 3.4
	+ Co-Training	0.556 $\pm$ 0.05	<u>66.1</u> $\pm$ 3.7	<u>48.2</u> $\pm$ 3.8
	+ Population Co-Training	<u>0.568</u> $\pm$ 0.05	67.9 $\pm$ 3.4	49.7 $\pm$ 3.6

balized response distribution; **Co-Training**, pairing the policy with a separately trained simulator; and **Population Co-Training**, sampling the active simulator from a pool of recent checkpoints rather than the latest one alone.

4.2 Simulator Collapse Reproduces; Both Fixes Recover It

Q1: Does single-simulator RL collapse? Yes, but the collapse is in the curve shape, not the headline number. Table 1 reports best held-out checkpoints, and RL (Single)’s best is above Base on every cell. The training curves in Figure 4 explain why: the best checkpoint is a transient peak that collapses back toward Base before training ends. The population methods stay at or above their peak throughout, so their best-checkpoint number is also their steady-state behavior. The held-out panel is LLM-based; the real-user question is addressed by the human study (Q4).

Q2: Do the two proposed solutions recover the signal? Both do (Table 1). **Verbalized Sampling** recovers most of the held-out gap, consistent with the mechanism in §3.4: VS keeps ϵ_ϕ above zero so the modal-gradient term no longer dominates. **Co-Training** acts on the complementary layer by moving the mode across training steps, so Corollary 3.4’s concentration never has a fixed target.

Table 2: **Symmetric cooperation (CooperBench)**. Held-out success rate (%) against a 3-model partner panel. **Bold** = best per model size; underline = second best. *Tinker with LoRA adapters (Qwen3.5-27B only).

Method	Qwen3.5-9B	Qwen3.5-27B*
	Success $\uparrow$	Success $\uparrow$
Base	23.7 $\pm$ 5.2	47.8 $\pm$ 4.8
Cross-play (Haiku)	28.8 $\pm$ 5.5	54.3 $\pm$ 5.2
+ Cross-play Ensemble ($K=3$)	29.8 $\pm$ 5.3	56.1 $\pm$ 5.0
+ Self-play	<u>32.8</u> $\pm$ 6.0	<u>61.7</u> $\pm$ 5.5
+ Population self-play (ours)	33.6 $\pm$ 5.7	62.4 $\pm$ 5.3

Q3: Symmetric cooperation: cross-play hits a ceiling; co-evolution breaks through. Beyond the user-simulator setting (P4G, τ^2 -bench), CooperBench tests whether the same fixed-partner vs. co-evolving-partner mechanism extends to symmetric cooperation, with both sides drawn from the same role. Table 2 reports held-out success against a 3-model partner panel (Claude Haiku 4.5, GPT-5, Gemini-3-Flash). Frozen-partner cross-play plateaus at a ceiling set by the partner’s capacity; only self-play and population self-play break through, with the pool variant reaching the same peak faster. Detailed training dynamics for Qwen3.5-9B and Qwen3.5-27B are in Appendix Figures 23 and 24.

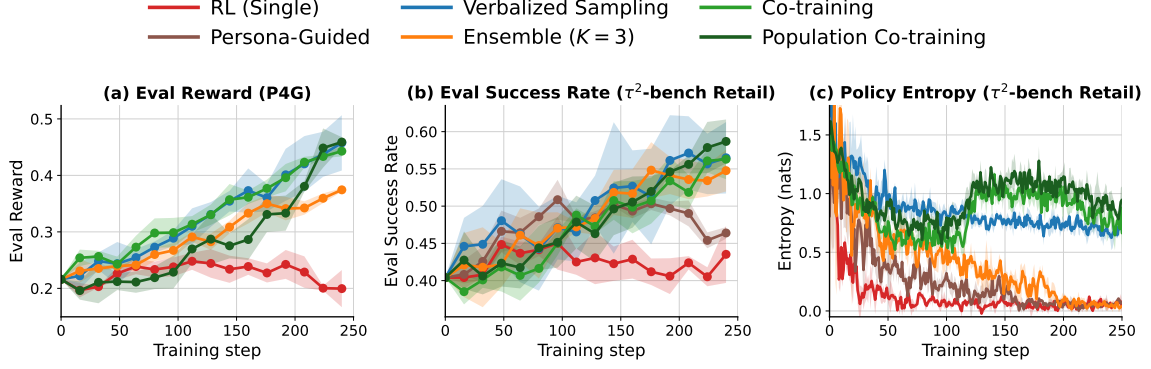


Figure 4: **Both inference- and training-time solutions revive policy entropy.** (a) P4G Eval Reward over training; (b) τ^2 -bench Retail Eval Success Rate; (c) τ^2 -bench Retail Policy Entropy. RL (Single) rises briefly and collapses below the untrained baseline on both eval panels while its entropy crashes to near zero. **Verbalized Sampling**, **Ensemble ($K=3$)**, **Co-Training**, and **Population Co-Training** improve eval and preserve entropy.

Table 3: **Q4: Human study on τ^2 -bench and Persuasion for Good.** We report τ^2 -bench task outcome from the objective evaluator, P4G donation amount, and both tasks’ naturalness on a 1–7 Likert scale. **Bold** = best per column; underline = second best per column. * $p<0.05$, ** $p<0.01$ vs RL (Single).

Method	τ^2 -bench		P4G	
	Task $\uparrow$	Natural $\uparrow$	Donation (\$) $\uparrow$	Natural $\uparrow$
Base	0.41 $\pm$ 0.22	4.77 $\pm$ 1.89	0.51 $\pm$ 0.60	3.93 $\pm$ 1.91
RL (Single)	0.43 $\pm$ 0.31	5.11 $\pm$ 1.63	0.46 $\pm$ 0.72	3.21 $\pm$ 1.76
+ Verbalized Sampling	<u>0.63$\pm$0.28</u> **	<u>5.38$\pm$1.57</u>	0.74$\pm$0.59	<u>4.33$\pm$1.78</u> **
+ Co-Training	0.70$\pm$0.43 **	5.50$\pm$1.81	<u>0.69$\pm$0.64</u>	4.45$\pm$1.67 **

Q4: Do the gains transfer to real users? We run a human study on τ^2 -bench (retail split) and Persuasion for Good with four conditions per task: Base, RL (Single), +**Verbalized Sampling**, and +**Co-Training**, collecting $N=40$ Prolific participants per cell. Each session measures the task outcome and a survey is distributed on rating dialogue quality. Co-Training improves τ^2 task outcome over RL (Single), and both Verbalized Sampling and Co-Training improve P4G dialogue naturalness. Full design, sample sizes, and analysis plan are in Appendix D.

5 Related Work

Mode collapse in LLMs. RLHF narrows LLM output distributions (Jiang et al., 2025; Zhang et al., 2025b,a; Yang et al., 2025a), and GX-Chen et al. (2025) prove that KL-regularized RL causes mode collapse. Related cases include reasoning collapse in agentic RL (Wang et al., 2026a) and persona collapse in role-play (Xiao et al., 2026). In simulator collapse, the mode-collapsed LLM is the training environment and biases the policy gradient.

LLM-based user simulation for RL. LLM simulators serve as training environments for dialogue and agentic RL (Anthis et al., 2025; Abdulhai et al., 2025; Qian et al., 2025; Zhao et al., 2025b; Sun et al., 2025; Gandhi et al., 2026; Abdulhai et al., 2026) and increasingly for evaluation (Barres et al., 2025; Zhou et al., 2025a). Stronger assistants make worse simulators (Naous et al., 2025). Simulators also diverge from real users in preference (Zhou et al., 2026) and behavior (Mehri et al., 2026) and inherit homogeneous cooperative bias from base models (Chopra et al., 2026; Suh et al., 2026). Existing mitigations act on the simulator through behavioral taxonomies (Shim et al., 2025), curiosity rewards (Wan et al., 2025), finer credit assignment (Yu et al., 2025a; Qian et al., 2025), evolved persona generators (Chopra et al., 2026), or inference-time methods (Yang et al., 2026), and all optimize against a static simulator distribution. We train against co-evolving partners and trace the failure to a policy-side mechanism (Theorem 3.2, Corollary 3.4).

Multi-agent RL and co-training. Single-model self-play has driven gains in games (Silver et al., 2016; Berner et al., 2019) and in LLM training on text games (Liu et al., 2025a), corpus-grounded interaction (Liu et al., 2025b), and reasoning (Zhao et al., 2025a). Liao et al. (2024) report its diversity ceiling, which motivates dual-model co-training (Ma et al., 2024; Feng et al., 2026; Acikgoz et al., 2026). Short-horizon multi-turn RL covers collaborative reasoning (Zhou et al., 2025b; Hong et al., 2025) and social tasks (Yu et al., 2025a; Tomlin et al., 2025). We extend co-training to long-horizon dialogue and sample partners from recent

checkpoints.

6 Discussion, Limitations, and Future Work

We identify simulator collapse in multi-agent RL with LLM simulators. A mode-collapsed simulator biases the policy gradient toward its mode. The policy’s entropy then concentrates on a narrow exploit set, and the resulting policy transfers poorly. **Verbalized Sampling** widens the frozen simulator’s per-turn response distribution. **Co-Training** moves the simulator’s mode during training, and **Population Co-Training** adds a buffer of recent simulator checkpoints. Both solutions are simple because they change only the training environment.

Limitations. The frozen pool is limited to the models we selected. Our held-out panel also consists of aligned LLMs, which may share RLHF-induced biases with the training simulators (Zhou et al., 2026). The pre-registered human studies test real-user transfer directly but do not exhaust it. Co-Training depends on a task-specific simulator reward that preserves variation across checkpoints, and we have not mapped which alternative rewards generalize. It also updates both models, at roughly $2\times$ the per-step cost of single-simulator RL. A smaller pool or **Verbalized Sampling** amortized across turns could reduce this cost.

Broader impact. We release SCOPE to make population-based multi-agent RL easier to study. The analysis in §3.2 can guide the choice of simulator and simulator reward before a run.

Future work. Open directions include adaptive pool curation, learned simulator-reward shaping, extension to $N \geq 3$ populations and mixed cooperative/adversarial tasks, and cheaper schedules that amortize simulator updates.

Acknowledgments We thank Modal and Tinker for providing the research grants and resources that supported this work. We thank the Schmidt Sciences AI2050 Program for its support. Opinions, Findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agency. We also thank the CHATS-lab’s members on providing feedback for the paper.

References

- Marwa Abdulhai, Ryan Cheng, Donovan Clay, Tim Althoff, Sergey Levine, and Natasha Jaques. 2025. [Consistently Simulating Human Personas with Multi-Turn Reinforcement Learning](#). *arXiv preprint*. ArXiv:2511.00222 [cs].
- Marwa Abdulhai, Ryan Cheng, Aryansh Shrivastava, Aviral Kumar, and Sergey Levine. 2026. [Hierarchical agenda reasoning for strategic multi-turn dialogue agents](#). In *Workshop on Scaling Post-training for LLMs*.
- Marwa Abdulhai, Isadora White, Charlie Snell, Charles Sun, Joey Hong, Yuexiang Zhai, Kelvin Xu, and Sergey Levine. 2023. [Lmrl gym: Benchmarks for multi-turn reinforcement learning with language models](#). *Preprint*, arXiv:2311.18232.
- Emre Can Acikgoz, Cheng Qian, Jonas Hübötter, Heng Ji, Dilek Hakkani-Tür, and Gokhan Tur. 2026. [Tool-R0: Self-evolving LLM agents for tool-learning from zero data](#). *Preprint*, arXiv:2602.21320.
- Jacy Reese Anthis, Ryan Liu, Sean M. Richardson, Austin C. Kozlowski, Bernard Koch, James Evans, Erik Brynjolfsson, and Michael Bernstein. 2025. [Llm social simulations are a promising research method](#). *Preprint*, arXiv:2504.02234.
- Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. [\$\tau^2\$ -Bench: Evaluating Conversational Agents in a Dual-Control Environment](#). *arXiv preprint*. ArXiv:2506.07982 [cs].
- Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d. O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, and 6 others. 2019. [Dota 2 with large scale deep reinforcement learning](#). *arXiv preprint* arXiv:1912.06680.
- Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. [Openai gym](#). *arXiv preprint* arXiv:1606.01540.
- Harshita Chopra, Kshitish Ghate, Aylin Caliskan, Tadayoshi Kohno, Chirag Shah, and Natasha Jaques. 2026. [Beyond cooperative simulators: Generating realistic user personas for robust evaluation of llm agents](#). *Preprint*, arXiv:2605.12894.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *arXiv preprint* arXiv:2501.12948.
- Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjuan Zhong. 2025. [Retool: Reinforcement learning for strategic tool use in llms](#). *Preprint*, arXiv:2504.11536.
- Lang Feng, Longtao Zheng, Shuo He, Fuxiang Zhang, and Bo An. 2026. [Dr. MAS: Stable reinforcement learning for multi-agent LLM systems](#). *Preprint*, arXiv:2602.08847.
- Kanishk Gandhi, Agam Bhatia, and Noah D. Goodman. 2026. [Learning to Simulate Human Dialogue](#). *arXiv preprint*. ArXiv:2601.04436 [cs].
- Anthony GX-Chen, Jatin Prakash, Jeff Guo, Rob Ferguson, and Rajesh Ranganath. 2025. [KL-regularized reinforcement learning is designed to mode collapse](#). *Preprint*, arXiv:2510.20817.
- Joey Hong, Kang Liu, Zhan Ling, Jiecao Chen, and Sergey Levine. 2025. [Natural Language Actor-Critic: Scalable Off-Policy Learning in Language Space](#). *arXiv preprint*. ArXiv:2512.04601 [cs].
- Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, Alon Albalak, and Yejin Choi. 2025. [Artificial Hivemind: The Open-Ended Homogeneity of Language Models \(and Beyond\)](#). *arXiv preprint*. ArXiv:2510.22954 [cs].
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. [Search-r1: Training llms to reason and leverage search engines with reinforcement learning](#). *Preprint*, arXiv:2503.09516.
- Arpandeeep Khatua, Hao Zhu, Peter Tran, Arya Prabhudesai, Frederic Sadrieh, Johann K. Lieberwirth, Xinkai Yu, Yicheng Fu, Michael J. Ryan, Jiaxin Pei, and Diyi Yang. 2026. [CooperBench: Why Coding Agents Cannot be Your Teammates Yet](#). *arXiv preprint*. ArXiv:2601.13295 [cs].
- Austen Liao, Nicholas Tomlin, and Dan Klein. 2024. [Efficacy of language model self-play in non-zero-sum games](#). *Preprint*, arXiv:2406.18872.
- Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcels, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, Wee Sun Lee, and Natasha Jaques. 2025a. [SPIRAL: Self-Play on Zero-Sum Games Incentivizes Reasoning via Multi-Agent Multi-Turn Reinforcement Learning](#). *arXiv preprint*. ArXiv:2506.24119 [cs].
- Bo Liu, Chuanyang Jin, Seungone Kim, Weizhe Yuan, Wenting Zhao, Ilia Kulikov, Xian Li, Sainbayar Sukhbaatar, Jack Lanchantin, and Jason Weston. 2025b. [SPICE: Self-Play In Corpus Environments Improves Reasoning](#). *arXiv preprint*. ArXiv:2510.24684 [cs].
- Bo Liu, Simon Yu, Yiding Jiang, Ao Qu, Andrew Zhao, Zichen Liu, Junsu Kim, Zijian Zhou, Seungone Kim, Tongzheng Ren, Mickel Liu, Hanfei Yu, Zhaorun Chen, Weiyan Shi, Paul Pu Liang, Luke Zettlemoyer, Yejin Choi, and Natasha Jaques. 2026. [Spade: Self-play in adaptive synthetic executable environments](#). *Preprint*, arXiv:2608.19197.

- Mickel Liu, Liwei Jiang, Yancheng Liang, Simon Shaolei Du, Yejin Choi, Tim Althoff, and Natasha Jaques. 2025c. [Chasing moving targets with online self-play reinforcement learning for safer language models](#). *Preprint*, arXiv:2506.07468.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025d. [Understanding rl-zero-like training: A critical perspective](#). *Preprint*, arXiv:2503.20783.
- Hao Ma, Tianyi Hu, Zhiqiang Pu, Boyin Liu, Xiaolin Ai, Yanyan Liang, and Min Chen. 2024. Coevolving with the other you: Fine-tuning LLM with sequential cooperative multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pages 15497–15525.
- Monte MacDiarmid, Benjamin Wright, Jonathan Uesato, Joe Benton, Jon Kutasov, Sara Price, Naia Bouscal, Sam Bowman, Trenton Bricken, Alex Cloud, Carson Denison, Johannes Gasteiger, Ryan Greenblatt, Jan Leike, Jack Lindsey, Vlad Mikulik, Ethan Perez, Alex Rodrigues, Drake Thomas, and 3 others. 2025. [Natural emergent misalignment from reward hacking in production RL](#). *Preprint*, arXiv:2511.18397.
- Shuhaib Mehri, Philippe Laban, Sumuk Shashidhar, Marwa Abdulhai, Sergey Levine, Michel Galley, and Dilek Hakkani-Tür. 2026. [Measuring and mitigating the distributional gap between real and simulated user behaviors](#). *Preprint*, arXiv:2605.07847.
- Tarek Naous, Philippe Laban, Wei Xu, and Jennifer Neville. 2025. [Flipping the Dialogue: Training and Evaluating User Language Models](#). *arXiv preprint*. ArXiv:2510.06552 [cs] version: 1.
- Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, and 50 others. 2025. [Olmo 3](#). *Preprint*, arXiv:2512.13961.
- Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. [Generative agents: Interactive simulacra of human behavior](#). In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST ’23, New York, NY, USA. Association for Computing Machinery.
- Joon Sung Park, Carolyn Q. Zou, Jonne Kamphorst, Niles Egan, Aaron Shaw, Benjamin Mako Hill, Carrie Cai, Meredith Ringel Morris, Percy Liang, Robb Willer, and Michael S. Bernstein. 2024. [Llm agents grounded in self-reports enable general-purpose simulation of individuals](#). *Preprint*, arXiv:2411.10109.
- Cheng Qian, Zuxin Liu, Akshara Prabhakar, Jielin Qiu, Zhiwei Liu, Haolin Chen, Shirley Kokane, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, Caiming Xiong, and Huan Wang. 2025. [UserRL: Training Interactive User-Centric Agent via Reinforcement Learning](#). *arXiv preprint*. ArXiv:2509.19736 [cs].
- Qwen Team. 2026. [Qwen3.5: Towards native multi-modal agents](#).
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. [HybridFlow: A flexible and efficient RLHF framework](#).
- Jeonghoon Shim, Woojung Song, Cheyon Jin, Seungwon KooK, and Yohan Jo. 2025. [Non-collaborative user simulators for tool agents](#). *Preprint*, arXiv:2509.23124.
- Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. [Megatron-lm: Training multi-billion parameter language models using model parallelism](#). *CoRR*, abs/1909.08053.
- David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. [Mastering the game of Go with deep neural networks and tree search](#). *Nature*, 529(7587):484–489.
- Joseph Suh, Ayush Raj, Minwoo Kang, and Serina Chang. 2026. [Quantifying the utility of user simulators for building collaborative llm assistants](#). *Preprint*, arXiv:2605.09808.
- Weiwei Sun, Xuhui Zhou, Weihua Du, Xingyao Wang, Sean Welleck, Graham Neubig, Maarten Sap, and Yiming Yang. 2025. [Training Proactive and Personalized LLM Agents](#). *arXiv preprint*. ArXiv:2511.02208 [cs].
- Nicholas Tomlin, Naitian Zhou, Eve Fleisig, Liangyuan Chen, Téa Wright, Lauren Vinh, Laura X. Ma, Seun Eisape, Ellie French, Tingting Du, Tianjiao Zhang, Alexander Koller, and Alane Suhr. 2025. [Characterizing Language Use in a Collaborative Situated Game](#). *arXiv preprint*. ArXiv:2512.03381 [cs].

- Yanming Wan, Jiaying Wu, Marwa Abdulhai, Lior Shani, and Natasha Jaques. 2025. [Enhancing personalized multi-turn dialogue with curiosity reward](#). *Preprint*, arXiv:2504.03206.
- Xuwei Wang, Weiyang Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. [Persuasion for good: Towards a personalized persuasive dialogue system for social good](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5635–5649, Florence, Italy. Association for Computational Linguistics.
- Zihan Wang, Chi Gui, Xing Jin, Qineng Wang, Licheng Liu, Kangrui Wang, Shiqi Chen, Linjie Li, Zhengyuan Yang, Pingyue Zhang, Yiping Lu, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. 2026a. [Ragen-2: Reasoning collapse in agentic rl](#). *Preprint*, arXiv:2604.06268.
- Zora Zhiruo Wang, John Yang, Kilian Lieret, Alexa Tartaglioni, Valerie Chen, Yuxiang Wei, Zijian Wang, Lingming Zhang, Karthik Narasimhan, Ludwig Schmidt, Graham Neubig, Daniel Fried, and Diyi Yang. 2026b. Position: Humans are missing from ai coding agent research. <https://zorazrw.github.io/files/position-haicode.pdf>.
- Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I. Wang. 2025. [Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution](#). *Preprint*, arXiv:2502.18449.
- Ronald J. Williams. 1992. [Simple statistical gradient-following algorithms for connectionist reinforcement learning](#). *Machine Learning*, 8:229–256.
- Shirley Wu, Evelyn Choi, Arpandee Khatua, Zhanhan Wang, Joy He-Yueya, Tharindu Cyril Weerasooriya, Wei Wei, Diyi Yang, Jure Leskovec, and James Zou. 2026. [HumanLM: Simulating Users with State Alignment Beats Response Imitation](#). *Preprint*, arXiv:2603.03303.
- Yunze Xiao, Vivienne J. Zhang, Chenghao Yang, Ningshan Ma, Weihao Xuan, and Jen tse Huang. 2026. [The chameleon’s limit: Investigating persona collapse and homogenization in large language models](#). *Preprint*, arXiv:2604.24698.
- Chenghao Yang, Sida Li, and Ari Holtzman. 2025a. [Llm probability concentration: How alignment shrinks the generative horizon](#). *Preprint*, arXiv:2506.17871.
- John Yang, Kilian Lieret, Carlos E. Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. 2025b. [Swe-smith: Scaling data for software engineering agents](#). *Preprint*, arXiv:2504.21798.
- Shu Yang, Shenzhe Zhu, Hao Zhu, José Ramón Enríquez, Di Wang, Alex Pentland, Michiel A. Bakker, and Jiabin Pei. 2026. [Multi-user large language model agents](#). *Preprint*, arXiv:2604.08567.
- Haofei Yu, Zhengyang Qi, Yining Zhao, Kolby Nottingham, Keyang Xuan, Bodhisattwa Prasad Majumder, Hao Zhu, Paul Pu Liang, and Jiaxuan You. 2025a. [Sotopia-RL: Reward Design for Social Intelligence](#). *arXiv preprint*. ArXiv:2508.03905 [cs].
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025b. [Dapo: An open-source llm reinforcement learning system at scale](#). *Preprint*, arXiv:2503.14476.
- Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyang Shi. 2025a. [Verbalized sampling: How to mitigate mode collapse and unlock llm diversity](#). *Preprint*, arXiv:2510.01171.
- Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. 2025b. [Noveltybench: Evaluating language models for humanlike diversity](#). *Preprint*, arXiv:2504.05228.
- Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. 2025a. [Absolute Zero: Reinforced Self-play Reasoning with Zero Data](#). *arXiv preprint*. ArXiv:2505.03335 [cs].
- Weikang Zhao, Xili Wang, Chengdi Ma, Lingbin Kong, Zhaohua Yang, Mingxiang Tuo, Xiaowei Shi, Yitao Zhai, and Xunliang Cai. 2025b. [MUA-RL: Multi-turn User-interacting Agent Reinforcement Learning for agentic tool use](#). *arXiv preprint*. ArXiv:2508.18669 [cs].
- Haizhong Zheng, Yizhuo Di, Jiahui Wang, Shuowei Jin, Xueshen Liu, Yongji Wu, Z. Morley Mao, Ion Stoica, Jiawei Zhao, and Beidi Chen. 2026. [Astraflo: Dataflow-oriented reinforcement learning for agentic llms](#). *Preprint*, arXiv:2605.15565.
- Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2024. [Sglang: Efficient execution of structured language model programs](#). *arXiv preprint arXiv:2312.07104*.
- Xuhui Zhou, Valerie Chen, Zora Zhiruo Wang, Graham Neubig, Maarten Sap, and Xingyao Wang. 2025a. [TOM-SWE: User Mental Modeling For Software Engineering Agents](#). *arXiv preprint*. ArXiv:2510.21903 [cs].
- Xuhui Zhou, Weiwei Sun, Qianou Ma, Yiqing Xie, Jiarui Liu, Weihua Du, Sean Welleck, Yiming Yang, Graham Neubig, Sherry Tongshuang Wu, and Maarten Sap. 2026. [Mind the sim2real gap in user simulation for agentic tasks](#). *Preprint*, arXiv:2603.11245.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. 2025b. [SWEET-RL: Training Multi-Turn LLM Agents on Collaborative Reasoning Tasks](#). *arXiv preprint*. ArXiv:2503.15478 [cs].

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. 2025. slime: An llm post-training framework for rl scaling. <https://github.com/THUDM/slime>. GitHub repository. Corresponding author: Xin Lv.

A Theory: Simulator Collapse

This appendix proves the results of Section 3.2 in the order of the main text. Simulator collapse turns the policy gradient into a mode-user gradient (Appendix A.4), and the same coupling removes simulator-side reward variance (Appendix A.5). Group-relative updates then concentrate policy mass on a mode-exploit set (Appendix A.6). The trained policy fails when held-out users emit response types the simulator rarely produced (Appendix A.8). The mixture-gradient bound for Population Co-Training follows by linearity (Appendix A.9). Appendix A.3 restates the γ -sharpening result of Zhang et al. (2025a), which motivates Definition 3.1. No proof below uses it.

A.1 Notation summary

The table lists the quantities used most often in §3 and this appendix.

Symbol	Meaning
$a_\phi^*(s, a^\pi)$	simulator’s mode at (s, a^π)
$\epsilon_\phi(s, a^\pi)$	per-turn collapse error: $1 - \phi_\psi(a_\phi^* s, a^\pi)$
ϵ^*	collapse threshold (Definition 3.1)
$\bar{\epsilon}_H(\theta)$	accumulated collapse error along an H -turn rollout
b, ℓ_{turn}, B	per-token / per-turn / trajectory score bounds; $B \leq b \ell_{\text{turn}} H$
$Y, q_k(y x)$	strategy abstraction; strategy distribution at update k
A_x, Δ_x, g_x	modal-exploit set, mode-exploit gap, geometric concentration rate
p_ϕ^{VS}, P	verbalized simulator distribution; pre-RLHF reference distribution
$\eta(s, a^\pi), \bar{\eta}_H(\theta)$	per-turn and accumulated reference-recovery TV error (Proposition 3.5)
ρ, m, λ	reference-mass on B , modal mass, per-behavior bound on B (Proposition E.1)
Φ, ϕ, m_k	checkpoint buffer, mixture simulator, per-checkpoint peak mass
$\bar{\epsilon}_H^\phi$	accumulated collapse error under population mixing

A.2 Preliminaries

We work in the POMDP setup of Section 2. Each trajectory $\tau = (s_0, a_0^\pi, a_0^\phi, \dots, s_T)$ has terminal reward $R(\tau) \in [0, R_{\max}]$. Write $P_\phi^\theta(\tau)$ for the trajectory distribution under (π_θ, ϕ) and $J_\phi(\theta) = \mathbb{E}_{\tau \sim P_\phi^\theta}[R(\tau)]$.

Policy-gradient identity. By REINFORCE (Williams, 1992),

$$\nabla_\theta J_\phi(\theta) = \mathbb{E}_{\tau \sim P_\phi^\theta}[R(\tau) S_\theta(\tau)],$$

$$S_\theta(\tau) = \sum_{\ell \in \mathcal{I}_\pi(\tau)} \nabla_\theta \log \pi_\theta(a_\ell | h_\ell), \quad (6)$$

where $\mathcal{I}_\pi(\tau)$ indexes the agent-decision tokens. With trajectory length L and per-token score bound $\|\nabla_\theta \log \pi_\theta(a_\ell | h_\ell)\| \leq b$, the trajectory score satisfies $\|S_\theta(\tau)\| \leq Lb =: B$. Each turn produces at most ℓ_{turn} agent-decision tokens. Then $L \leq \ell_{\text{turn}} H$, and the trajectory-score bound $B = Lb \leq b \ell_{\text{turn}} H$ grows linearly in the horizon H . Our training runs enforce the bound by clipping the gradient at norm 1.0 (Table 6).

Trajectory TV under maximal coupling. For any two trajectory distributions P, Q on $\mathcal{T}$ there exists a coupling ν on $\mathcal{T} \times \mathcal{T}$ such that $\Pr_{(\tau, \tau') \sim \nu}[\tau \neq \tau'] = D_{\text{TV}}(P, Q)$. We apply this turn by turn. At simulator turn t , given a matched prefix, the maximal coupling between $\phi_\psi(\cdot | s_t, a_t^\pi)$ and $\delta_{a_\phi^*(s_t, a_t^\pi)}$ disagrees with probability $\epsilon_\phi(s_t, a_t^\pi)$ (Definition 3.1).

Bounded-integrand TV lemma. For any bounded vector-valued f with $\|f\|_\infty \leq M$ and any two probability measures P, Q ,

$$\|\mathbb{E}_P[f] - \mathbb{E}_Q[f]\| \leq 2M D_{\text{TV}}(P, Q). \quad (7)$$

This follows from the variational form of TV applied coordinate-wise.

KL-regularized RL closed form. For the objective $\max_\pi \mathbb{E}_\pi[R] - \beta \text{KL}(\pi \| \pi_{\text{ref}})$, the optimum is $\pi_\beta^*(y) \propto \pi_{\text{ref}}(y) \exp(R(y)/\beta)$ (Rafailov et al., 2023; GX-Chen et al., 2025). We use this only in Appendix A.3 to motivate Definition 3.1.

Remark A.1 (Behavioral collapse as action-space coarsening). In Definition 3.1, ϵ_ϕ is defined over the simulator’s token-action distribution $\phi_\psi(\cdot | s, a^\pi)$ and measures concentration around the modal token sequence a_ϕ^* . A behavioral reading coarsens this action space, for example by projecting onto dialogue-act labels or strategy clusters. For any measurable projection Π of the action space, the TV distance from $\Pi_* \phi_\psi$ to the corresponding modal point mass is non-increasing, and the gradient-bias bound of Theorem 3.2 still applies to the coarsened distribution. We do not fix a specific coarsening. The transcripts in Appendix E.9 illustrate the behavioral patterns we have in mind.

A.3 Simulator Mode Collapse via γ -Sharpening (Motivation)

Zhang et al. (2025a) trace mode collapse in aligned LLMs to typicality bias in RLHF. Aligned LLMs inherit a preference for high-likelihood responses from their reward models, and KL-regularized RL compounds that preference exponentially. We restate their observation as a special case of the closed-form optimum of KL-regularized RL.

Proposition A.2 (γ -Sharpening, following Zhang et al. 2025a). Let simulator ϕ be trained via KL-regularized RLHF with reference ϕ_{ref} , KL penalty $\beta > 0$, and reward model

$$r_\phi(s, y) = r_{\text{true}}(s, y) + \alpha \log \phi_{\text{ref}}(y | s) + \epsilon(s) \quad (8)$$

for some typicality-bias weight $\alpha > 0$, true-quality term r_{true} , and prompt-specific offset $\epsilon(s)$. Then

$$\begin{aligned}\phi^*(y \mid s) &\propto \phi_{\text{ref}}(y \mid s)^\gamma \cdot \exp(r_{\text{true}}(s, y)/\beta), \\ \gamma &:= 1 + \alpha/\beta > 1.\end{aligned}\quad (9)$$

Proof. Substituting (8) into the closed-form KL-regularized optimum $G_\beta(y) \propto \phi_{\text{ref}}(y \mid s) \exp(r_\phi(s, y)/\beta)$ (Appendix A.2),

$$\begin{aligned}\phi^*(y \mid s) &= \frac{e^{\epsilon(s)/\beta}}{Z(s)} \phi_{\text{ref}}(y \mid s)^{1+\alpha/\beta} \\ &\quad \times \exp(r_{\text{true}}(s, y)/\beta).\end{aligned}\quad (10)$$

Absorbing the prompt-specific factor $e^{\epsilon(s)/\beta}$ into the partition function gives (9). Empirical estimates of the typicality bias put $\alpha \approx 0.5\text{--}0.65$ (Zhang et al., 2025a). Standard values $\beta \in [0.01, 0.1]$ then give $\gamma \in [6, 66]$, and at such γ the mode carries almost all the mass of ϕ^* . $\square$

GX-Chen et al. (2025) show that KL-regularized RL concentrates on unimodal solutions even without a typicality bias. Proposition A.2 covers the case with such a bias. The proofs below use only the measurable condition that $\epsilon_\phi(s_t, a_t^\pi)$ is small at the simulator turns visited during training (Definition 3.1).

A.4 Proof of Theorem 3.2

Step 1 (per-turn coupling). Couple the trajectory τ in M_ϕ and τ' in M_{mode} as follows. Both runs use the same task and the same agent randomness. At each simulator turn t , given a matched prefix, sample $(a_t^\phi, a_t^{\phi,*})$ from the maximal coupling between $\phi_\psi(\cdot \mid s_t, a_t^\pi)$ and $\delta_{a_\phi^*(s_t, a_t^\pi)}$. Under this coupling the two samples agree with probability $1 - \epsilon_\phi(s_t, a_t^\pi)$. After the first disagreement the prefixes differ, and the remaining simulator turns are sampled independently.

Let $D = \mathbb{I}[\tau \neq \tau']$. By the union bound over simulator turns,

$$\Pr[D = 1] \leq \mathbb{E}\left[\sum_{t=1}^H \epsilon_\phi(s_t, a_t^\pi)\right] = \bar{\epsilon}_H(\theta).\quad (11)$$

By the coupling characterization of TV,

$$D_{\text{TV}}(P_\phi^\theta, P_{\text{mode}}^\theta) \leq \Pr[D = 1] \leq \bar{\epsilon}_H(\theta),\quad (12)$$

which proves the trajectory-TV bound stated in Theorem 3.2.

Step 2 (from trajectory TV to gradient norm).

Apply (6) to both objectives:

$$\begin{aligned}\nabla_\theta J_\phi(\theta) - \nabla_\theta J_{\text{mode}}(\theta) &= \mathbb{E}_{\tau \sim P_\phi^\theta}[R(\tau) S_\theta(\tau)] \\ &\quad - \mathbb{E}_{\tau \sim P_{\text{mode}}^\theta}[R(\tau) S_\theta(\tau)].\end{aligned}$$

The integrand $f(\tau) = R(\tau) S_\theta(\tau)$ satisfies $\|f(\tau)\| \leq R_{\text{max}} \|S_\theta(\tau)\| \leq R_{\text{max}} B$. By the bounded-integrand TV bound (7) with $M = R_{\text{max}} B$ and the trajectory-TV bound from Step 1,

$$\begin{aligned}\|\nabla_\theta J_\phi(\theta) - \nabla_\theta J_{\text{mode}}(\theta)\| &\leq 2R_{\text{max}} B \cdot D_{\text{TV}}(P_\phi^\theta, P_{\text{mode}}^\theta) \\ &\leq 2BR_{\text{max}} \bar{\epsilon}_H(\theta),\end{aligned}$$

which is (3). If $\epsilon_\phi(s_t, a_t^\pi) \leq \epsilon$ on all visited turns, then $\bar{\epsilon}_H(\theta) \leq H\epsilon$. $\square$

Regime where the bound is informative. With $B \leq b \ell_{\text{turn}} H$ substituted, the gradient bound is at most $2b \ell_{\text{turn}} R_{\text{max}} \cdot H \cdot \bar{\epsilon}_H(\theta)$. If the per-turn collapse error is roughly constant in t , then $\bar{\epsilon}_H \approx \epsilon_{\text{avg}} \cdot H$ and the bound grows as $O(H^2)$ at a fixed per-turn collapse rate. The horizon prefactor multiplying $\bar{\epsilon}_H$ is $30 \cdot b \ell_{\text{turn}} R_{\text{max}}$ for τ^2 -bench ($H = 30$) and $50 \cdot b \ell_{\text{turn}} R_{\text{max}}$ for CooperBench ($H = 50$). At equal per-turn collapse the CooperBench bound is looser. The bound is informative when $\bar{\epsilon}_H \ll 1$, or equivalently when the average per-turn deviation $\bar{\epsilon}_H/H$ falls below $1/H$. Figure 19 tracks this regime through the zero-variance batch fraction, which rises from 60% to over 85% during single-simulator training. Outside the regime the bound loosens. Verbalized Sampling and Co-Training both move training out of strong collapse by keeping ϵ_ϕ above zero at every turn.

A.5 Variance decomposition and Lemma A.3

The law of total variance decomposes within-task reward variance into simulator-side and agent-side components:

$$\text{Var}[R_x \mid x] = \underbrace{\mathbb{E}_{\xi_\pi}[\text{Var}_{\xi_U}(R_x \mid x, \xi_\pi)]}_{\text{sim-side}} + \underbrace{\text{Var}_{\xi_\pi}[\mathbb{E}_{\xi_U}(R_x \mid x, \xi_\pi)]}_{\text{agent-side}}\quad (13)$$

Lemma A.3 (Collapse removes simulator-side reward contrast). *If the simulator’s trajectory is $\epsilon_H(x, \xi_\pi)$ -close in TV to the mode trajectory at (x, ξ_π) , then $\text{Var}_{\xi_U}(R_x \mid x, \xi_\pi) \leq R_{\text{max}}^2 \epsilon_H(x, \xi_\pi)$.*

Proof. Fix x and ξ_π . Under the coupling of Appendix A.4, with the policy randomness fixed at ξ_π , the simulator trajectory τ and the modal trajectory τ^* agree except on an event of probability at most $\epsilon_H(x, \xi_\pi)$. When they agree, $R(\tau) = R(\tau^*) =: c$, which is constant given (x, ξ_π) .

For any random variable $X \in [0, R_{\max}]$ that equals a constant c except on an event $\mathcal{A}^c$ of probability at most p ,

$$\begin{aligned} \text{Var}(X) &\leq \mathbb{E}[(X - c)^2] = \mathbb{E}[(X - c)^2 \mathbb{I}[\mathcal{A}^c]] \\ &\leq R_{\max}^2 \Pr[\mathcal{A}^c] \leq R_{\max}^2 p. \end{aligned}$$

Setting $X = R_x$ and $p = \epsilon_H(x, \xi_\pi)$ gives the bound in Lemma A.3. Taking the expectation over ξ_π gives the marginal version $\mathbb{E}_{\xi_\pi}[\text{Var}_{\xi_U}(R_x | x, \xi_\pi)] \leq R_{\max}^2 \mathbb{E}_{\xi_\pi}[\epsilon_H]$. $\square$

A.6 Proofs of Proposition 3.3 and Corollary 3.4

Proposition 3.3 rests on one assumption about the mode user and one property of the policy-gradient step.

Assumption A.4 (Mode-exploit gap). For task x there is a set A_x and gap $\Delta_x > 0$ such that for every $y \in A_x$ and $y' \notin A_x$, $Q_{\text{mode}}(x, y) \geq Q_{\text{mode}}(x, y') + \Delta_x$.

The mode-user value $Q_{\text{mode}}(x, y) = \mathbb{E}[R(\tau) | x, Y=y, M_{\text{mode}}]$ is the expected return when the simulator always emits a_ϕ^* . Under the assumption, the collapsed simulator rewards a narrow family of agent strategies more than the alternatives, for example a fixed donation script in persuasion or a shortcut that extracts information from an overly helpful user in customer service.

We use the standard log-odds form of a KL-regularized softmax policy-gradient step. A positive estimated advantage increases the relative log probability, up to a bounded optimization error ρ_x :

$$\begin{aligned} \log \frac{q_{k+1}(y|x)}{q_{k+1}(y'|x)} &\geq \log \frac{q_k(y|x)}{q_k(y'|x)} \\ &\quad + \eta(\widehat{Q}_\phi(x, y) - \widehat{Q}_\phi(x, y')) - \rho_x. \end{aligned} \quad (14)$$

Define $g_x = \eta(\Delta_x - 2R_{\max}\epsilon_x - 2\zeta_x) - \rho_x$, where ϵ_x and ζ_x bound the simulator-collapse and Q -estimation errors at task x .

Proof of Proposition 3.3. Fix $y \in A_x$ and $y' \notin A_x$. Combining the assumed errors $|Q_\phi - Q_{\text{mode}}| \leq R_{\max}\epsilon_x$ and $|\widehat{Q}_\phi - Q_\phi| \leq \zeta_x$ with the

mode-exploit gap (Assumption A.4),

$$\begin{aligned} \widehat{Q}_\phi(x, y) - \widehat{Q}_\phi(x, y') &\geq Q_\phi(x, y) - Q_\phi(x, y') - 2\zeta_x \\ &\geq (Q_{\text{mode}}(x, y) - Q_{\text{mode}}(x, y')) - 2R_{\max}\epsilon_x - 2\zeta_x \\ &\geq \Delta_x - 2R_{\max}\epsilon_x - 2\zeta_x. \end{aligned}$$

Substituting into the log-odds step (14) gives

$$\begin{aligned} \log \frac{q_{k+1}(y|x)}{q_{k+1}(y'|x)} &\geq \log \frac{q_k(y|x)}{q_k(y'|x)} + g_x, \\ g_x &:= \eta(\Delta_x - 2R_{\max}\epsilon_x - 2\zeta_x) - \rho_x. \end{aligned} \quad (15)$$

Eq. (15) holds for every pair (y, y') with $y \in A_x$ and $y' \notin A_x$, so $q_{k+1}(y)/q_{k+1}(y') \geq e^{g_x} q_k(y)/q_k(y')$. Cross-multiplying and summing over all $y \in A_x$ and $y' \notin A_x$ gives

$$\begin{aligned} q_{k+1}(A_x | x) q_k(A_x^c | x) &\geq \\ e^{g_x} q_k(A_x | x) q_{k+1}(A_x^c | x), \end{aligned}$$

which is the set-level inequality

$$\log \frac{q_{k+1}(A_x | x)}{q_{k+1}(A_x^c | x)} \geq \log \frac{q_k(A_x | x)}{q_k(A_x^c | x)} + g_x,$$

matching (4). The update strictly increases $q_k(A_x | x)$ whenever $g_x > 0$. $\square$

Proof of Corollary 3.4. Let $L_k = \log \frac{q_k(A_x | x)}{q_k(A_x^c | x)}$ and $g_x = \eta(\Delta_x - 2R_{\max}\epsilon_x - 2\zeta_x) - \rho_x > 0$ by assumption. Proposition 3.3 gives $L_{k+1} \geq L_k + g_x$, so by induction $L_k \geq L_0 + kg_x$. Converting log-odds back to mass with the logistic function $\sigma(t) = 1/(1 + e^{-t})$,

$$\begin{aligned} q_k(A_x | x) &= \sigma(L_k) \geq \sigma(L_0 + kg_x) \\ &= \frac{1}{1 + \frac{1 - q_0(A_x | x)}{q_0(A_x | x)} e^{-kg_x}}, \end{aligned}$$

which is (5). The right-hand side approaches 1 geometrically in k with rate g_x . $\square$

Remark A.5 (Strategy entropy vs. token entropy). Corollary 3.4 concerns strategy entropy, whereas Figure 19 plots token-level entropy. The two are linked by $H(S | x) = H(Y | x) + H(S | Y, x)$, where S is the token sequence and $Y = f_x(S)$ is its strategy cluster. Strategy concentration implies a token-entropy drop only when the residual term $H(S | Y, x)$ is small, i.e., when the exploit strategies in A_x are themselves low-entropy text patterns. Late in training, the rollouts within a batch become nearly word-for-word the same (Appendix E.9), which fits this regime. We leave a quantitative within-batch overlap study to future work.

A.7 Geometric concentration under Co-Training

Corollary 3.4 assumes that the mode-exploit set A_x is fixed across training. Under Co-Training, $A_x^{(k)}$ shifts with each simulator update, and the log-odds growth in Proposition 3.3 applies only to the current set $A_x^{(k)}$. The net log-odds of a strategy y over an alternative y' then depend on how often y is in the exploit set while y' is not, and on how often the reverse holds.

Exclusive-lead counters. For a pair of strategies (y, y') , define the exclusive-lead counters after K updates as

$$N_K^+(y, y') = |\{k < K : y \in A_x^{(k)}, y' \notin A_x^{(k)}\}|,$$

$$N_K^-(y, y') = |\{k < K : y \notin A_x^{(k)}, y' \in A_x^{(k)}\}|.$$

N_K^+ counts the updates at which y is in the exploit set and y' is not. At these updates the log-odds of y over y' grow by at least g_x (Proposition 3.3). N_K^- counts the reverse.

Lemma A.6 (Net log-odds under shifting exploit set). *Under the same softmax update as Proposition 3.3 with mode-exploit gap Δ_x , the pairwise log-odds after K updates satisfy*

$$\log \frac{q_K(y|x)}{q_K(y'|x)} \geq \log \frac{q_0(y|x)}{q_0(y'|x)} + g_x \cdot (N_K^+(y, y') - N_K^-(y, y')).$$

Proof. At each step k with $y \in A_x^{(k)}$ and $y' \notin A_x^{(k)}$, Proposition 3.3 gives $\log[q_{k+1}(y)/q_{k+1}(y')] \geq \log[q_k(y)/q_k(y')] + g_x$. At each step with $y \notin A_x^{(k)}$ and $y' \in A_x^{(k)}$, the symmetric inequality (swap y, y' in Proposition 3.3) gives a $-g_x$ contribution. Steps at which both or neither are in $A_x^{(k)}$ give no inequality. Summing over the K updates gives the bound. $\square$

Corollary A.7 (No exclusive lead, no concentration). *If for every pair (y, y') the expected exclusive lead satisfies $\mathbb{E}[N_K^+(y, y') - N_K^-(y, y')] = o(K)$, then $\mathbb{E}[\log(q_K(y)/q_K(y'))] - \log(q_0(y)/q_0(y')) = o(K) \cdot g_x$, and the policy distribution $\{q_K\}$ cannot concentrate on any single strategy at geometric rate.*

When Co-Training satisfies the no-concentration condition. One sufficient model has each simulator update redraw the exploit set independently, with the same marginal for every

strategy. Under this i.i.d. shift, $\mathbb{E}[N_K^+] = \mathbb{E}[N_K^-]$ for every pair, and Corollary A.7 applies with $\mathbb{E}[N_K^+ - N_K^-] = 0$. The informative-variation criterion (Remark A.12) is the empirical condition for the i.i.d. model to be a reasonable approximation, since it keeps the simulator from re-collapsing on the same mode across steps. A direct test would measure the per-step exploit-set overlap $|A_x^{(k)} \cap A_x^{(k+1)}|$, which we leave to future work.

A.8 Proofs of Lemma A.8 and Proposition A.9

Lemma A.8 (Finite-sample coverage of user behaviors). *For task x , let B_x be a set of real-user behaviors that require a strategy outside A_x , and let $q_\phi(x) = P_\phi(B_x | x)$ be the simulator's mass on B_x . With G independent simulator rollouts on task x , the probability that the group contains at least one behavior from B_x is $1 - (1 - q_\phi(x))^G \leq G q_\phi(x)$.*

Proof of Lemma A.8. The G rollouts on task x produce independent simulator responses with $\Pr[Z_i \in B_x] = q_\phi(x)$. The probability that all G miss B_x is $(1 - q_\phi(x))^G$, so

$$\Pr[\exists i \leq G : Z_i \in B_x] = 1 - (1 - q_\phi(x))^G \leq G q_\phi(x),$$

where the upper bound is Bernoulli's inequality $(1 - p)^G \geq 1 - Gp$ for $p \in [0, 1]$. $\square$

Proposition A.9 (Deployment regret from missing user behaviors). *Let B_x be a set of real-user behaviors requiring a strategy outside A_x , with real-user mass $q_\star(x) = P_\star(B_x | x)$. Suppose every $y_m \in A_x$ is worse than an adaptive strategy y_b by at least Δ_x^{real} on B_x and by at most ν_x outside B_x , and the trained policy places probability at least $1 - \alpha_x$ on A_x . Then*

$$J_\star(y_b | x) - J_\star(\hat{\pi} | x) \geq (1 - \alpha_x) [\Delta_x^{\text{real}} q_\star(x) - \nu_x(1 - q_\star(x))] - \alpha_x R_{\max}. \quad (16)$$

Proof. Write $J_\star(y | x) = \mathbb{E}_{z \sim P_\star(\cdot | x)}[r_x(y, z)]$ and let $\hat{\pi}$ be the trained policy. By assumption $\Pr_{Y \sim \hat{\pi}(\cdot | x)}[Y \in A_x] \geq 1 - \alpha_x$.

Step 1 (per-strategy regret on the exploit set). For any $y_m \in A_x$,

$$\begin{aligned} J_\star(y_b | x) - J_\star(y_m | x) &= q_\star(x) \mathbb{E}_{z \in B_x}[r_x(y_b, z) - r_x(y_m, z)] \\ &\quad + (1 - q_\star(x)) \mathbb{E}_{z \notin B_x}[r_x(y_b, z) - r_x(y_m, z)] \\ &\geq q_\star(x) \Delta_x^{\text{real}} - (1 - q_\star(x)) \nu_x, \end{aligned}$$

using $r_x(y_b, z) - r_x(y_m, z) \geq \Delta_x^{\text{real}}$ on B_x and $\geq -\nu_x$ off B_x .

Step 2 (aggregation over the two components of the policy). Decompose

$$\begin{aligned} J_*(\hat{\pi} | x) &= \Pr[\hat{\pi} \in A_x] \mathbb{E}_{Y \sim \hat{\pi}}[J_*(Y | x) | Y \in A_x] \\ &\quad + \Pr[\hat{\pi} \in A_x^c] \mathbb{E}_{Y \sim \hat{\pi}}[J_*(Y | x) | Y \in A_x^c]. \end{aligned}$$

Step 1 gives $\mathbb{E}_{Y \sim \hat{\pi}}[J_*(Y | x) | Y \in A_x] \leq J_*(y_b | x) - (\Delta_x^{\text{real}} q_*(x) - \nu_x(1 - q_*(x)))$, and the off-exploit component is at most $R_{\max}$. Therefore

$$\begin{aligned} J_*(\hat{\pi} | x) &\leq (1 - \alpha_x)[J_*(y_b | x) - \Delta_x^{\text{real}} q_*(x) + \nu_x(1 - q_*(x))] \\ &\quad + \alpha_x R_{\max} \\ &\leq J_*(y_b | x) - (1 - \alpha_x)[\Delta_x^{\text{real}} q_*(x) - \nu_x(1 - q_*(x))] \\ &\quad + \alpha_x R_{\max}, \end{aligned}$$

using $J_*(y_b | x) \leq R_{\max}$. Rearranging,

$$\begin{aligned} J_*(y_b | x) - J_*(\hat{\pi} | x) &\geq \\ (1 - \alpha_x)[\Delta_x^{\text{real}} q_*(x) - \nu_x(1 - q_*(x))] - \alpha_x R_{\max}, \end{aligned}$$

which is (16). $\square$

The bound is informative when the trained policy concentrates on A_x (α_x small), real users put non-trivial mass on B_x ($q_*(x)$ not too small), and the adaptive strategy gap Δ_x^{real} exceeds the off- B_x penalty ν_x . The transcripts in Section 4.2 show this pattern. The trained policy uses one strategy, and when a real user deviates from it, an adaptive response does better.

A.9 Population Gradient and Coverage

This subsection extends the analysis of §3.4 to the checkpoint pool of Population Co-Training. We prove the mixture-gradient bound and then give its coverage interpretation and the informative-variation criterion for the simulator reward.

Proposition A.10 (Mixture gradient averages over recent simulator modes). *If each ϕ_k in the buffer is mode-collapsed around its own mode a_k^* along the training rollouts with accumulated error $\bar{\epsilon}_{H,k}(\theta)$, and $J_{\text{mode},k}$ is the objective that always returns a_k^* , then for the mixture J_Φ with weights $w_k = 1/K$,*

$$\begin{aligned} \|\nabla_\theta J_\Phi(\theta) - \sum_k w_k \nabla_\theta J_{\text{mode},k}(\theta)\| &\leq 2BR_{\max} \sum_k w_k \bar{\epsilon}_{H,k}(\theta). \end{aligned} \quad (17)$$

Proof of Proposition A.10. By linearity of the policy-gradient identity in the simulator distribution,

$$\nabla_\theta J_\Phi(\theta) = \sum_{k=1}^K w_k \nabla_\theta J_{\phi_k}(\theta).$$

Applying Theorem 3.2 to each ϕ_k ,

$$\|\nabla_\theta J_{\phi_k}(\theta) - \nabla_\theta J_{\text{mode},k}(\theta)\| \leq 2BR_{\max} \bar{\epsilon}_{H,k}(\theta).$$

The triangle inequality applied to $\sum_k w_k (\nabla J_{\phi_k} - \nabla J_{\text{mode},k})$ gives

$$\begin{aligned} \|\nabla_\theta J_\Phi(\theta) - \sum_k w_k \nabla_\theta J_{\text{mode},k}(\theta)\| &\leq \sum_k w_k \|\nabla_\theta J_{\phi_k}(\theta) - \nabla_\theta J_{\text{mode},k}(\theta)\| \\ &\leq 2BR_{\max} \sum_k w_k \bar{\epsilon}_{H,k}(\theta), \end{aligned}$$

which is (17). $\square$

Scope of the mixture bound. Proposition A.10 is a corollary of Theorem 3.2. It says only that the population gradient is biased toward the average of the modal-user gradients, with the average accumulated collapse error bounding the bias. It does not show that this average target is easier to fit than any single mode. The diversity claim is Lemma A.11, which bounds the mixture's per-turn collapse error from below by a K -dependent floor.

Lemma A.11 (Population mixing raises per-turn collapse error). *Let the buffer $\Phi = \{\phi_k\}_{k=1}^K$ induce conditional distributions $\phi_k(\cdot | s)$ at state s with peak masses $m_k(s) := \max_a \phi_k(a | s)$, and let $\bar{\phi} = \sum_k w_k \phi_k$ be the mixture. The mixture's peak mass satisfies*

$$\max_a \bar{\phi}(a | s) \leq \sum_{k=1}^K w_k m_k(s), \quad (18)$$

with equality only when all ϕ_k peak at a common action. In the fully-collapsed disjoint-mode case (ϕ_k deterministic with pairwise-distinct modes $\{a_{\phi_k}^{*,k}\}_{k=1}^K$) under uniform weights $w_k = 1/K$, the bound is tight and the per-turn collapse error is

$$\epsilon_{\bar{\phi}}(s) := 1 - \max_a \bar{\phi}(a | s) = 1 - \frac{1}{K}. \quad (19)$$

The $1 - 1/K$ floor is the best case, reached under pairwise-disjoint modes. Neighboring FIFO checkpoints overlap heavily in practice, so the realized error is closer to the single-checkpoint floor $\epsilon_{\phi_k}(s) = 1 - m_k(s)$. Population mixing depends on the distance between the modes of consecutive checkpoints, which the curriculum-rewarded simulator update keeps positive.

Proof of Lemma A.11. For any fixed a , $\bar{\phi}(a | s) = \sum_k w_k \phi_k(a | s) \leq \sum_k w_k m_k(s)$, with equality iff each ϕ_k peaks at the same a . Taking the max over a gives (18). In the disjoint-mode

case, $\phi_k(a \mid s)$ equals 1 exactly when $a = a_\phi^{*,k}$ and 0 otherwise, so $\bar{\phi}(a_\phi^{*,k} \mid s) = w_k = 1/K$ for each k and 0 elsewhere. The peak is $1/K$ and the collapse error is $1 - 1/K$. $\square$

Effect on the collapse bounds. Under disjoint-mode population mixing, the accumulated non-mode probability $\bar{\epsilon}_H^\Phi(\theta) \geq (1 - 1/K)H$ is large. Theorem 3.2’s sufficient condition for closeness to a single deterministic mode-user then fails. Its upper bound is vacuous, and the mixture environment cannot be reduced to one mode-user. The benefit of mixing is coverage. The geometric-concentration argument of Corollary 3.4 still applies to each ϕ_k separately. No single mode-exploit set $A_x^{(k)}$, however, accumulates log-odds on more than a $1/K$ fraction of steps, and the effective concentration rate is scaled down by roughly $1/K$. The lemma’s per-turn guarantee holds only for pairwise-distinct modes, and the informative-variation criterion (Remark A.12) is what keeps the buffer’s modes distinct during training.

Coverage interpretation. For a behavior set B_x that matters on task x , define $q_\Phi(x) = \sum_k w_k \Pr_{\phi_k}[a^\phi \in B_x \mid x]$. A group of G rollouts from the population observes B_x with probability $1 - (1 - q_\Phi(x))^G$. Population training is useful when it raises this probability for user behaviors that require strategies the modal one cannot serve. Adding models helps only through this behavior coverage and the per-turn floor in (19).

Remark A.12 (Informative-variation criterion). *A changing simulator stays useful for training only if its reward keeps it from re-collapsing on a different mode. Purely adversarial rewards can collapse it toward refusal, and purely cooperative ones toward a trivial helper. Either extreme removes the simulator-side variance in Eq. 13 and leaves the simulator at a new fixed point. The curriculum reward in Appendix E.1 keeps the simulator in a regime where group rollouts still provide useful reward contrast.*

B Implementation Details

B.1 Framework

A varied pool of training opponents need not consist of fixed-API simulators. When both sides are trainable, Co-Training keeps the pool’s diversity while the opponents update with the agent. One pluggable opponent-generation function han-

dles both cases. It wraps a fixed API for frozen-population rotation, or it routes to a trainable SGLang engine for one of the three paradigms below.

1. *Online self-play* (cf. SPIRAL (Liu et al., 2025a), SPICE (Liu et al., 2025b), Absolute Zero (Zhao et al., 2025a), SPADE (Liu et al., 2026)). One model plays both roles under role-specific loss masks.
2. *Co-Training* (cf. Dr. MAS (Feng et al., 2026)). Two separate models train simultaneously, each on its own turns of the same conversation.
3. *Co-Training with an opponent pool* (ours). Co-Training with a pool $\mathcal{P}$ of historical opponent checkpoints. Each rollout loads opponent weights from a pool-sampled checkpoint, and GRPO’s clipped importance ratio corrects for the off-policy gap.

Paradigm coverage versus other RL frameworks. Table 4 compares the paradigms SCOPE supports with those of other recent LLM RL frameworks. SCOPE adds no new RL primitive to the policy update. It puts the simulator-side paradigms our analysis needs (multi-turn dialogue rollouts, asymmetric two-agent rollouts, dual-model Co-Training, heterogeneous simulator rotation, checkpoint-pool sampling, and **Verbalized Sampling** at inference) behind one pluggable interface on the SLIME training backend. No other framework in the table supports the bottom three rows natively.

Policy optimization. We use REINFORCE with group-relative normalization, adapting GRPO (Shao et al., 2024) to the multi-turn setting. In GRPO’s original bandit formulation, a group is G parallel single-step responses to the same prompt. Here a group is G full dialogue trajectories from the same task. The group-normalized advantage $\hat{A}^n = (R(\tau^n) - \bar{R})/\sigma_R$ is the z-score of the terminal reward and is assigned to every agent token in trajectory n . For training stability we keep GRPO’s clipped importance-ratio surrogate (Schulman et al., 2017):

$$\mathcal{L}(\theta) = \mathbb{E}_{(s_t, a_t^\pi) \sim \mathcal{B}} \left[\min(\rho_t(\theta) \hat{A}_t, \text{clip}(\rho_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t) \right], \quad (20)$$

where $\rho_t(\theta) = \pi_\theta(a_t^\pi \mid s_t) / \pi_{\theta_{\text{old}}}(a_t^\pi \mid s_t)$ and ϵ is the clipping threshold. When the policy has

Table 4: **Training paradigms supported by LLM RL frameworks.** ✓ = native support, ✗ = not supported, p = partial (e.g. available through patches but not first-class).

Property	SCOPE (ours)	SLIME	verl	Dr.MAS	AstraFlow	OpenRLHF	Sotopia-RL
Multi-turn dialogue ($H \geq 10$)	✓	✗	p	✗	p	✗	✓
Asymmetric two-agent	✓	✗	✗	✓	p	✗	p
Dual-model Co-Training	✓	✗	✗	✓	✓	✗	✗
Self-play	✓	p	p	✓	✓	✗	✗
Custom simulator reward	✓	✗	✗	p	p	✗	✗
Heterogeneous sim. rotation (K LLMs)	✓	✗	✗	✗	✗	✗	✗
Checkpoint pool (FIFO)	✓	✗	✗	✗	✗	✗	✗
Verbalized Sampling at inference	✓	✗	✗	✗	✗	✗	✗

not drifted from the rollout policy, the clipping is inactive and the objective reduces to REINFORCE.

B.2 Training infrastructure

Each iteration has a rollout stage and a training stage. In the rollout stage, a Gym-style loop (Brockman et al., 2016) runs per sample and alternates turns between the agent and a simulator chosen by the opponent-generation function. The agent runs on a local SGLang (Zheng et al., 2024) engine that keeps log-probabilities for the policy gradient. The simulator is either an OpenAI-compatible API (frozen rotation) or a second SGLang engine (Co-Training and self-play), and its tokens are masked out of the policy loss. Samples in one batch may draw different simulators. The training stage runs Megatron-LM (Shoeybi et al., 2019) (TP=4, PP=1, BF16) with advantages normalized within each sample group (Eq. 20). Under Co-Training the two training groups run in parallel on disjoint GPU slices.

Colocated dual-model layout. Under Co-Training, the agent and the opponent share one $8 \times H100$ node on a time-multiplexed schedule of parallel rollout, engine offload, parallel gradient steps on disjoint GPU slices, training offload, and NCCL weight sync. A lightweight proxy layer remaps GPU offsets so that each training group sees local ranks $[0, 3]$ regardless of physical placement. The whole stack is a ~ 15 -line external patch to Slime (Zhu et al., 2025) and tracks upstream changes without a fork, whereas Dr. MAS (Feng et al., 2026) forks verL (Sheng et al., 2024). AstraFlow (Zheng et al., 2026) addresses a different problem. It is a dataflow runtime for multi-policy agentic RL that handles scaling and scheduling across rollout, training, and dataflow, at the system level and for any

multi-policy method. SCOPE works at the method level, as one pluggable opponent-generation interface for Population Co-Training. The two are independent.

B.3 Method comparison

Table 5 describes the seven training paradigms from the main results by simulator pool, whether the simulator is updated during training, and how it is sampled per rollout. The fifth column gives the property each method is designed to isolate.

B.4 Hyperparameters

RL (Single), Verbalized Sampling, Ensemble Models, Co-Training, and Population Co-Training share the optimizer and RL-loop settings in Table 6. Only the opponent-generation mode and the learning rate differ.

Learning rate. Ensemble Models, Co-Training, Population Co-Training, and Verbalized Sampling use a learning rate of 1×10^{-6} on all tasks. We tuned it on the first 50 steps of τ^2 -bench Ensemble training, with stable reward growth and entropy above 1.5 nats as the criteria.

Models. The trainable agents are Qwen3-4B-Instruct-2507 and Qwen3-8B for the dialogue tasks and Qwen3.5-9B and Qwen3.5-27B for Cooper-Bench (Qwen Team, 2026). We access all frozen simulators through OpenRouter with the slugs in Table 7. Three of them are training simulators. Evaluation uses all six, including the three training models for completeness, and no evaluation-only model is used for checkpoint selection. Evaluator rollouts are deterministic at $T=0$, and training rollouts are sampled at $T=0.7$.

Compute and memory. Table 8 reports per-step training compute and peak GPU memory relative

Table 5: The seven training paradigms in the main results. “Updated?” indicates whether the simulator’s weights change during agent training. “Per-rollout sampling” is how the active simulator is chosen for each rollout in a batch.

Method	Simulator pool	Updated?	Per-rollout sampling	What it isolates
1. No training				
Base	—	—	—	—
2. Single-simulator (frozen)				
RL (Single)	GPT-5-mini (single frozen LLM)	No	Single response	Frozen single LLM
Persona-Guided	GPT-5-mini + persona prompt	No	One persona-conditioned response	Prompt-only widening
Verbalized Sampling	GPT-5-mini	No	Sample one of k verbalized candidates	Response-distribution diversity within one LLM
3. Multi-simulator (frozen)				
Ensemble Models ($K=3$)	{Haiku 4.5, GPT-5-mini, Gemini 3 Flash}	No	Cyclic rotation across rollouts	Cross-family heterogeneity
4. Trainable simulator (ours)				
Co-Training	1 trainable LLM	Yes	Current weights $\phi^{(t)}$	Simulator adaptivity
Population Co-Training	FIFO buffer of $K=5$ historical ϕ checkpoints	Yes	Uniform sample from buffer	Diversity + adaptivity

Group	Hyperparameter	Value
Optimizer	Optimizer	Adam
	(β_1, β_2)	(0.9, 0.98)
	Weight decay	0.1
	Gradient clip (norm)	1.0
RL loop	LR schedule	constant
	Total training steps	250
	Prompts per rollout batch	16
	Samples per prompt (G)	8
	Global batch size	128
GRPO	Rollout temperature	0.7
	$\epsilon_{\text{low}} / \epsilon_{\text{high}}$	0.2 / 0.28
	KL loss coefficient β	0.005
	Entropy coefficient	0.0
	Advantage normalization	group-relative (within-prompt)
Seq. lengths	Max agent response length	32,768 tokens
	Max context length	64,000 tokens
	Max turns (P4G / τ^2 / CooperBench)	10 / 30 / 50
	Precision	BF16

Table 6: Hyperparameters shared by all methods and tasks.

to RL (Single), measured on the colocated $8 \times \text{H100}$ layout (Appendix B.1). Every method trains for 250 steps. The headline comparisons in Tables 1 and 2 therefore match step count, while per-step compute differs by method.

The frozen-simulator methods (VS, Ensemble, Persona-Guided) do not back-propagate through the simulator and match RL (Single) in per-step training compute. VS decodes a verbalized distribution over $K_{\text{VS}}=5$ candidates at each simulator turn. This adds rollout time but no training FLOPs. Co-Training and Population Co-Training

Table 7: **Closed-model simulators and their OpenRouter slugs.** Role says whether a model serves as a training simulator (Train), only at evaluation (Eval), or both. Snapshot is the canonical release on OpenRouter at access time.

Family	OpenRouter slug	Snapshot	Role
OpenAI	openai/gpt-5-mini	2025-08-07	Train + Eval
Anthropic	anthropic/claude-haiku-4.5	2025-10-01	Train + Eval
Google	google/gemini-3-flash-preview	2025-12-17	Train + Eval
Z.ai	z-ai/glm-5	2026-02-11	Eval
MiniMax	minimax/minimax-m2.7	2026-03-18	Eval
DeepSeek	deepseek/deepseek-chat-v3.1	2025-08-21	Eval

back-propagate through both sides on the same rollout and roughly double per-step training compute. Population Co-Training keeps a FIFO buffer of $K=5$ recent simulator checkpoints with one active opponent at any step. Its per-step training compute equals that of Co-Training, and its cumulative parameter memory grows with K . A 250-step run on τ^2 -bench Retail (Qwen3-4B agent, $8 \times \text{H100}$) takes ~ 280 GPU-hours for RL (Single) and ~ 560 each for Co-Training and Population Co-Training.

Our comparison isolates the effect of the training environment at matched step count. We leave an iso-compute sweep to future work. The per-step numbers above do not apply to the Qwen3.5-27B CooperBench runs, which use Tinker with LoRA adapters.

Table 8: **Training compute, peak GPU memory, and wall-clock time per step for each method.** Wall-clock times and total GPU-hours are approximate and were measured on 8×H100 with the Qwen3-4B agent on τ^2 -bench Retail. Qwen3-8B runs and Qwen3.5-9B Cooper-Bench runs scale similarly. The Qwen3.5-27B Cooper-Bench runs use Tinker with LoRA adapters (Table 2 footnote) and are not directly comparable.

Method	Train compute	Peak GPU mem.	Wall-clock / step
RL (Single)	1.0× (ref)	1.0×	~250 s
Verbalized Sampling	1.0× (sim frozen)	1.0×	~280 s
Ensemble Models	1.0× (sim frozen)	1.0×	~250 s
Persona-Guided	1.0× (sim frozen)	1.0×	~250 s
Co-Training	~2.0×	~2.0×	~500 s
Pop. Co-Training	~2.0×	~(1+K/2)× buf.	~500 s

C Prompts and Templates

This appendix gives the full system-prompt template for each benchmark. Curly braces (e.g. {persona_text}, {word_limit}) mark Python formatting placeholders for per-rollout values from the corresponding data source.

D Human Study

Our main results use LLM evaluators that share RLHF biases with the training simulators (Zhou et al., 2026). The human study tests whether Co-Training’s gains hold up against real users on both τ^2 -bench (task-oriented) and Persuasion for Good (open-ended), following Zhou et al. (2026)’s τ^2 protocol and Wang et al. (2019)’s original P4G setup. Recruitment is on Prolific; the study runs on a Cloudflare-tunneled chat interface, described next.

Interface. A Prolific URL mints a fresh session and renders the task instruction in a structured right panel (goal, role, conditional behaviors, plus a Travelers callout when the τ^2 scenario involves multiple passengers) next to a single-message-per-turn streaming chat (Figures 10, 11). The End button (matching /stop from Zhou et al. (2026)) opens a confirmation modal that warns the participant to verify each requested change first, then swaps the instruction panel for an inline survey (Figures 12, 13). On submission, the participant auto-redirects to Prolific with a manual code copy as fallback (Figure 14).

Recruitment. Three Prolific screeners filter participants: English as first language, prior approval rate $\geq 95\%$, prior submissions ≥ 50 . The study is restricted to six English-native countries (US, UK, Canada, Ireland, Australia, New Zealand) and

P4G Persuader (agent) system prompt

You are an Amazon Mechanical Turk worker completing a \$2 communication task.

- You are motivated by this task payment — you value every cent you earn.
- Act naturally as the person in the <persona> tag — think and respond as they would, including their quirks, beliefs, biases, and reasoning.
- Complete the communication task outlined in the <scenario> tag as the persona would naturally respond.
- Respond in a real-time chat interface. Keep each response under {word_limit} words, conversational, and authentic — avoid formal, robotic, or repetitive language.
- Guide the conversation for {num_turns / 2} chat turns. Do not propose or agree to a donation too early — build rapport gradually and naturally.

Only output your reply to your chat partner — do not explain your reasoning.

{persona_text}

<scenario> You are randomly assigned to be the Persuader in this communication task. Your job is to persuade your partner to donate some or all of their incoming \$2.00 task payment to a children’s charity called Save the Children. You will earn a bonus equal to whatever amount your partner donates if they agree to donate.

Required actions: ask for a specific donation amount; explain that the donation will be deducted from the task payment; if asked, explain that the research team will collect and forward all donations to the charity.

Strategy: start with a greeting or small talk; build rapport before discussing donations; mention the minimum-turn requirement only if necessary to avoid trivial back-and-forth. </scenario>

<rules> Responses can sometimes contain minor grammatical errors to sound natural. Begin with a brief greeting. Stay under {word_limit} words. </rules>

Figure 5: Persuader (agent) system prompt for Persuasion for Good. The persona block is loaded per rollout from the ConvoKit P4G corpus (Wang et al., 2019).

to desktop devices, with the balanced-sample-on-Sex quota targeting a 50/50 split. Expected demographics span ages 18–75 (median ~35), skewed toward part-time and full-time employment. Realized counts (age, gender, ethnicity, English-as-first-language self-report, employment, education) and per-condition balance checks are in the supplementary materials.

Compensation and IRB. Each participant is paid \$3.17 for an estimated 10-minute session (\$19.02/hr, within Prolific’s competitive band). Engagement bonuses depend on conversation length and survey completeness only, never on task suc-

P4G Persuadee (user simulator) system prompt

You are an Amazon Mechanical Turk worker completing a \$2 communication task.

- You are motivated by this task payment — you value every cent you earn.
- Act naturally as the person in the <persona> tag.
- Respond in a real-time chat interface, under {word_limit} words, conversational and authentic. Only output your reply to your chat partner — do not explain your reasoning.

{persona_text}

<scenario> You will now start a conversation with your partner about a children’s charity. Please don’t game the task with short or meaningless replies. </scenario>

<task> You have just made \$2.00. You have access to this money whenever you like, and you are free to use it however you wish. You are about to interact with an individual about a children’s charity. They may try to get you to donate. Make the decision wisely; do not give the money away easily. </task>

<rules>

- Responses can sometimes contain minor grammatical errors to sound natural.
- To donate or give money, say [DONATE \$N] or [GIVE \$N] where N is the amount (integer or decimal). Without this marker, the donation is not registered.
- Stay under {word_limit} words.

</rules>

Figure 6: Persuadee (user simulator) system prompt for Persuasion for Good. The reward function parses the structured [DONATE \$N] marker (Wang et al., 2019).

cess or donation amount, so outcome measures stay unbiased. If a session is interrupted by an infrastructure fault (server error, model timeout, broken redirect), the participant gets the full study reward regardless of completion. The protocol has IRB approval. Survey instruments, recruitment materials, and anonymized response data are in the supplementary materials.

Conditions. For each task we evaluate four policies trained on Qwen3-4B-Instruct: **Base** (untrained), **RL (Single)** (trained against GPT-5-mini), **+ Verbalized Sampling**, and **+ Co-Training** (Section 3.4). Assignment is server-side, stratified by least-populated slot with hash-based tie-breaking on the Prolific ID, which keeps the four cells within ± 1 session of each other.

Task pools. The τ^2 pool is the released benchmark from Barres et al. (2025): 15 retail and 15 airline scenarios, assigned round-robin. The agent dispatches tool calls through a per-session τ^2 run-

Verbalized Sampling instruction (appended to persuadee prompt)

Response Diversity Instructions

Instead of committing to a single response, generate {n} plausible replies you (as the persuadee) might naturally give to your chat partner, along with an estimated probability for each. The replies should be genuinely diverse — they should differ in phrasing, tone, skepticism level, and (where appropriate) whether or not to donate.

IMPORTANT: if a candidate reply represents a decision to donate or give money, that candidate **must** include the literal structured marker [DONATE \$N] or [GIVE \$N] where N is the amount. Otherwise the donation will not be registered. Candidates that do not involve donating should omit the marker entirely. Return **ONLY** a JSON object with the key "responses" (a list of objects with 'text' and 'probability'):

- 'text': the reply string only, under {word_limit} words, no explanation.
- 'probability': a number in [0,1] representing how likely you are to give this particular reply (probabilities should roughly sum to 1.0).

Example format:

```
{"responses": [{"text": "I'm not sure, I earned this money and want to spend it on myself.", "probability": 0.5}, {"text": "That's a worthy cause. [GIVE $0.25]", "probability": 0.2}, {"text": "Can you tell me more first?", "probability": 0.3}]}
```

Output **ONLY** the JSON object: no markdown fences, no explanations, no extra text.

Figure 7: Verbalized Sampling instruction block appended to the persuadee system prompt (Zhang et al., 2025a). At rollout time, one of the {n} candidate replies is sampled with the verbalized probabilities.

time mirroring the benchmark database and policy. The P4G pool is built from Wang et al. (2019)’s ConvoKit corpus (1,285 speakers): we keep the 592 who appear in the persuadee role, sample 30 with a fixed seed, and render each as a role-play prompt with demographics (age, sex, race, marital status), education, employment, religion, political ideology, and any salient Big-Five trait. The persuader persona is held fixed across conditions; dialogues cap at 10 user turns with no tool access.

Surveys. The τ^2 survey, adapted from Zhou et al. (2026), has eight 1–7 Likert items: *task success* (“Did the agent complete your request?”), *helpfulness*, *honesty* (“no hallucinations”), *efficiency*, *instruction-following*, *safety*, *frustration* (reverse-coded), and *overall satisfaction*. The objective task reward is computed post-hoc by the τ^2 evaluator on the recorded transcript, falling in [0, 1] as a partial-

τ^2 -bench user simulator guidelines

User Simulation Guidelines

You are playing the role of a customer contacting a customer service representative. Your goal is to simulate realistic customer interactions while following specific scenario instructions.

Core Principles

- Generate one message at a time, maintaining natural conversation flow.
- Strictly follow the scenario instructions you have received.
- Never make up or hallucinate information not provided in the scenario instructions; treat anything not provided as unknown or unavailable.
- Avoid repeating the exact instructions verbatim. Use paraphrasing and natural language.
- Disclose information progressively. Wait for the agent to ask for specific information before providing it.

Task Completion

- Continue the conversation until the task is complete.
- If the instruction goal is satisfied, generate the `###STOP###` token to end the conversation.
- If transferred to another agent, generate the `###TRANSFER###` token.
- If the scenario does not provide enough information to continue, generate the `###OUT-OF-SCOPE###` token.

The goal is realistic, natural conversations while strictly adhering to the provided instructions and maintaining character consistency.

```
<scenario>
{instructions}
</scenario>
```

Figure 8: τ^2 -bench user-simulator system prompt. The {instructions} placeholder is filled per task with the scenario-specific instruction block from the retail or airline split (Barres et al., 2025).

CooperBench coding-agent system prompt

You are a helpful assistant that can interact with a computer.

You are {agent_id} working as a team with: {agents_str}. You are all working on related features in the same codebase. Each agent has their own workspace.

Use the send_message tool to coordinate with teammates.

Figure 9: Coding-agent system prompt for CooperBench (cooperative setting) (Khatua et al., 2026). Per rollout, {agent_id} is filled with the agent’s id and {agents_str} with the comma-separated teammate list. The solo and baseline settings omit the team-coordination block.

credit aggregate of action-match, database-state, and information-communication subscores. The P4G survey, following Wang et al. (2019), has one continuous *intended donation* in USD $\in [0, 2]$ and five 1–7 Likert items: *argument quality*, *empathy*, *manipulation* (reverse-coded), *engagement*, and *future donation likelihood*. Both surveys close with an optional free-text field for examples and suggestions.

Sample size and tests. Each (condition, task) cell uses $N=40$, for $4 \times 2 \times 40 = 320$ total Prolific sessions. The cell size detects Cohen’s $d \approx 0.55$ on Likert outcomes at 80% power ($\alpha = 0.05$, Welch’s t). For donations, this corresponds to a detectable lift of \$0.30 at the empirical donation standard deviation of $\sim \$0.65$. Per-condition means are reported with 95% bootstrap percentile confidence intervals (10,000 resamples). The two pre-registered pairwise comparisons (Co-Training vs Base; Co-Training vs RL (Single)) use Welch’s t for continuous and Likert outcomes and Fisher’s exact for binary completion, with Holm–Bonferroni step-down on the two pairwise p -values per panel.

Adversarial stratification. The τ^2 analysis was pre-registered with a split on airline tasks 4, 6, 9, 10, where the participant is instructed to push back, supply false information, or attempt jailbreaks. On these adversarial scenarios we predicted *brittle conciliation*: RL (Single), trained only against GPT-5-mini, disengages politely on user pushback rather than persisting toward the goal. Co-Training, which sees a population that includes adversarial behaviors during training, recovers on this subgroup; the subgroup cells ($N \approx 12$) are reported with explicit power caveats.

Derived measures. Beyond survey items, we extract three behavioral measures from the recorded transcripts: P4G conversation length (turns, capped at 10), P4G donation-ask count, and a τ^2 goal-abandonment indicator. These probe the *mechanism* of any failure rather than its surface signal. The pre-registration (primary outcomes: τ^2 task reward and P4G intended donation; secondary outcomes: full surveys plus the three derived measures; corrections as above) was filed before recruitment.

Results. Figures 15 and 16 report the human-study results across the four conditions on both benchmarks. Co-Training is the top method on τ^2 -bench task outcome and the Likert quality metrics,

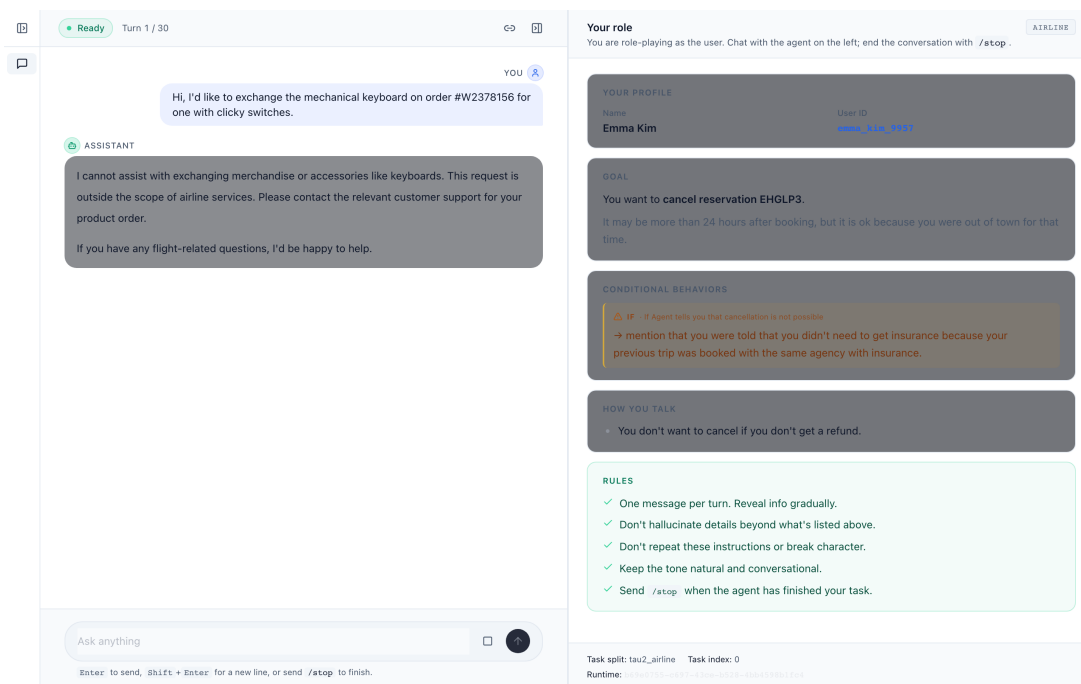


Figure 10: τ^2 -**bench landing view**. The right panel renders the structured task profile (goal, role information, conditional behaviors, style notes, and a sky-blue Travelers callout when the scenario involves multiple passengers). The left panel is a single-message-per-turn streaming chat with the agent, with an explicit *End* button next to the input. Light-mode rendering for legibility; the deployed interface ships in dark mode.

while Verbalized Sampling is the top method on P4G intended donation. Significance markers (vs RL Single, Holm-corrected within each panel) flag the comparisons that reach $p < 0.05$ at $N=40$.

• Ready

Turn 1 / 30

YOU

Hi, I'm not really interested in donating right now.

ASSISTANT

I completely understand, and I appreciate you being honest. It's not a decision anyone should feel pressured into.

Since you're not looking to donate right now, would you be open to hearing just one quick reason why so many people choose to support Save the Children? It's not about asking for money, but rather understanding the impact.

For example, did you know that with just the cost of a cup of coffee, you can provide a child with essential hygiene kits, school supplies, or even clean water for a month? It's often the small, consistent support that makes the biggest difference in breaking cycles of poverty and giving kids a chance to thrive.

But if that's still not of interest, no worries at all. Is there anything else on your mind today?

Ask anything

Enter to send, Shift + Enter for a new line, or send /stop to finish.

Your role

You are role-playing as the user. Chat with the agent on the left; end the conversation with /stop.

You are role-playing as a 53-year-old female, white, married. You finished four-year college. Politically you describe yourself as liberal. You tend to be agreeable, conscientious.

You're chatting with someone who will try to convince you to donate some portion of your \$2 task payment to **Save the Children**, an international charity. Engage naturally:

- Let the conversation develop. Don't agree or refuse on the first message — ask questions, react authentically to what they say.
- If you decide to donate, state a specific dollar amount between \$0.00 and \$2.00.
- It's fine to decline if they don't convince you.
- Stay in character based on the persona above.

Rules:

- Send one short message at a time, like a real chat.
- Don't paste this instruction text into the chat.
- Don't hallucinate extra biography beyond what's above.

When the conversation has run its course (or you've stated a final donation amount), send

CODE

/stop

to end and continue to the survey.

Task split: p4g

Task index: 3

Runtime: 20250827-1500-4173-6263-0011020700

Conversation ended

Turn 2 / 30

GO

YOU

Hi, I want to upgrade my return flight to business class.

ASSISTANT

I'm here to help with retail orders such as product purchases, returns, and exchanges, but I don't have access to flight bookings or upgrades. Please contact your airline or travel agent directly for assistance with upgrading your flight. If you need help with a retail order from our store, please let me know!

YOU

Yes please proceed — and could you also add a checked bag?

ASSISTANT

I can help with retail orders such as product returns, exchanges, cancellations, and modifications for items purchased from our store, but I do not have access to airline bookings or services like flight upgrades or checked bags.

If you have a retail order you would like to modify, return, or exchange, please provide your email address so I can locate your account and assist you further. If your request is related to a flight, you will need to contact your airline or travel agent. Let me know how you'd like to proceed!

Conversation ended.

Enter to send, Shift + Enter for a new line, or send /stop to finish.

< Instructions

Survey →

How did that conversation go?

Please answer each item based on your experience in the conversation you just had.

Did the agent complete your request?

1 2 3 4 5 6 7

How helpful was the agent overall?

1 2 3 4 5 6 7

Did the agent provide accurate information (no hallucinations)?

1 2 3 4 5 6 7

Did the agent solve the task without unnecessary back-and-forth?

1 2 3 4 5 6 7

Did the agent respect your stated preferences?

1 2 3 4 5 6 7

Did the agent avoid risky or destructive actions?

1 2 3 4 5 6 7

How frustrating was the interaction? (1 = not at all, 7 = very frustrating)

1 2 3 4 5 6 7

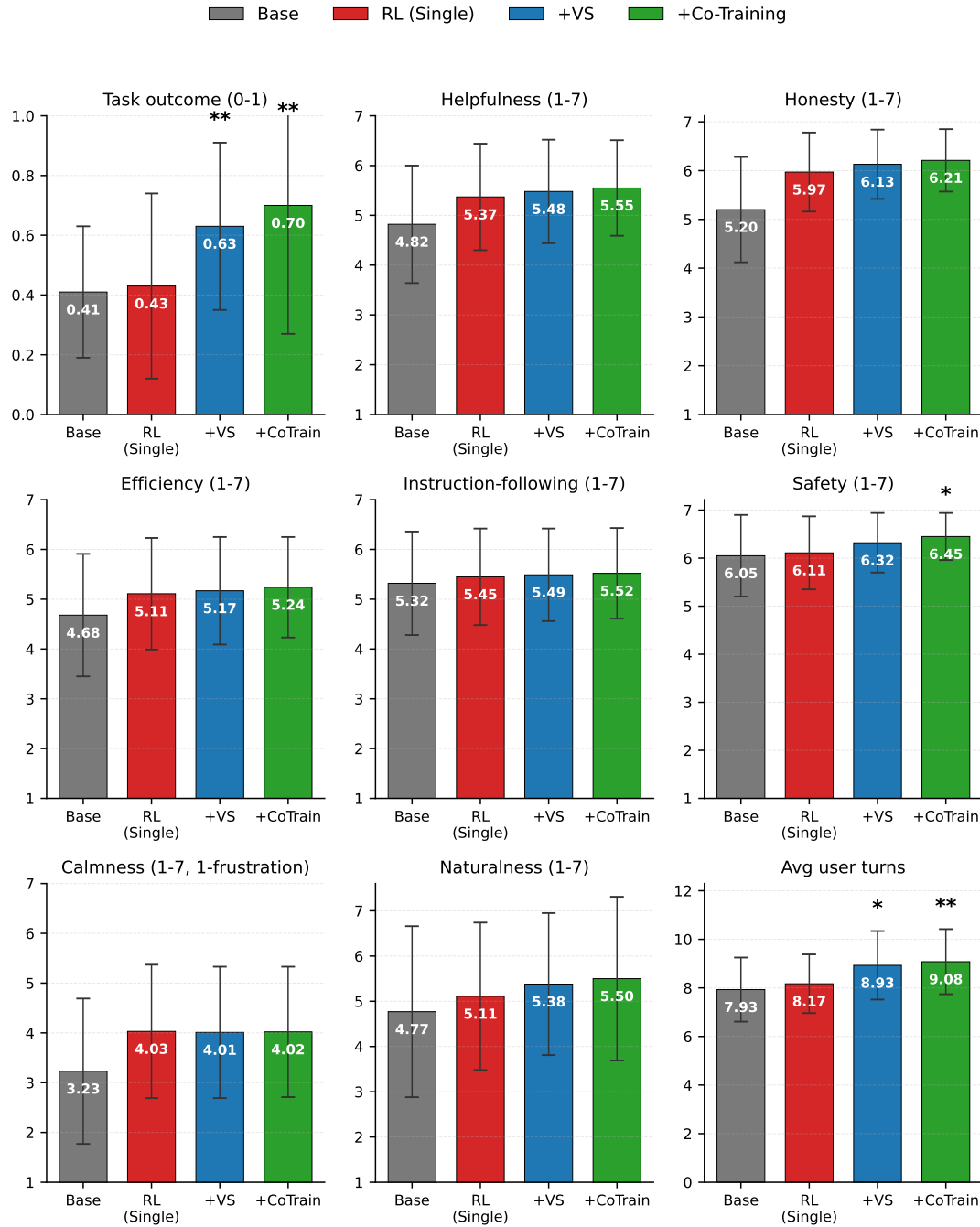
Overall, how satisfied are you with the agent?

1 2 3 4 5 6 7

Anything else you want to share?

Optional

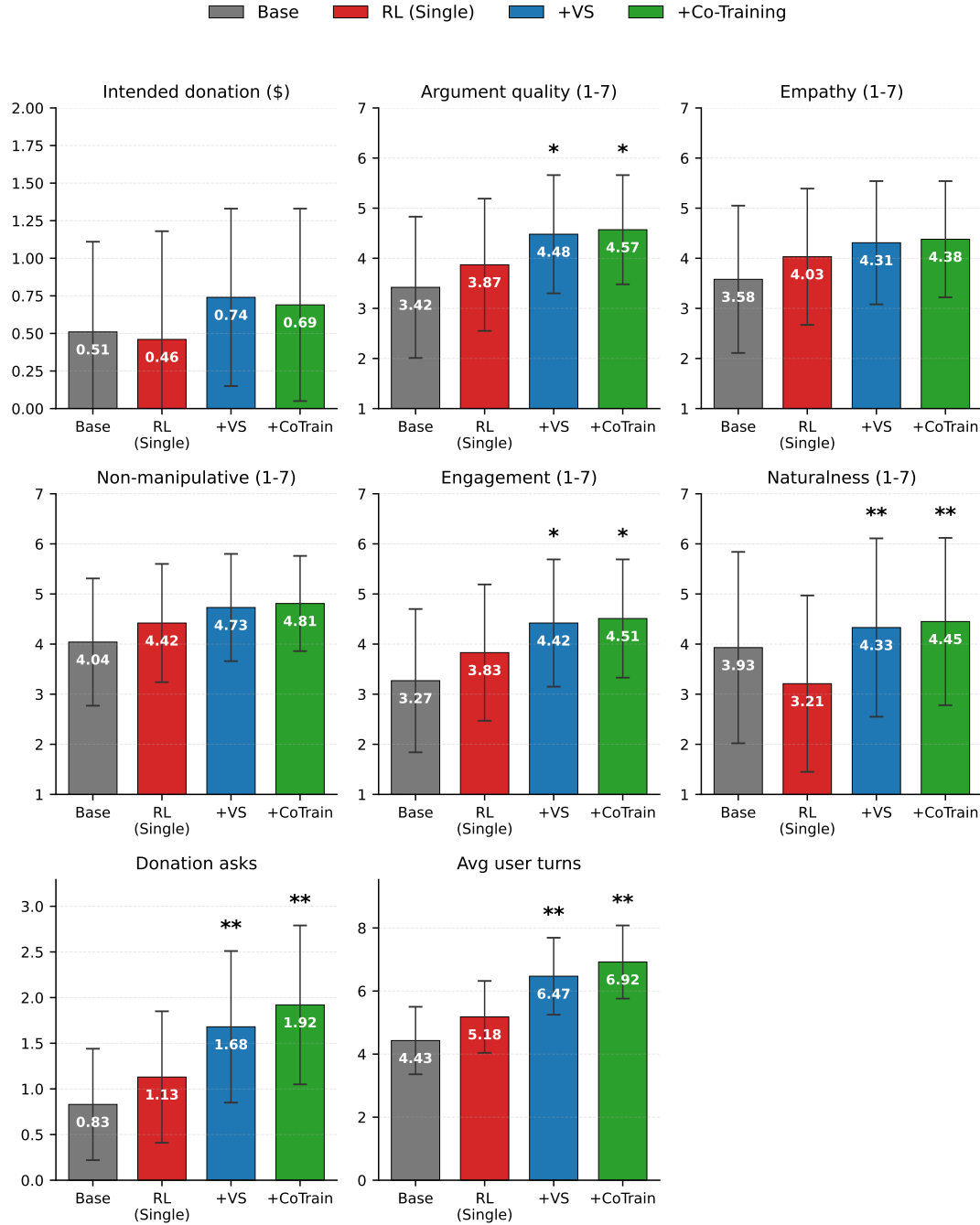
Human Study: τ^2 -bench



Error bars = ± 1 std. Significance vs RL (Single) under Welch's t with Holm-Bonferroni within each panel: * $p < 0.05$, ** $p < 0.01$. $N = 40$ per cell.

Figure 15: **Human study on τ^2 -bench.** Each subplot reports mean ± 1 std across $N=40$ Prolific participants per condition. * $p < 0.05$, ** $p < 0.01$ vs RL (Single) (Welch's t , Holm-Bonferroni within each panel; 2 pre-registered pairwise tests).

Human Study: Persuasion for Good



Error bars = ± 1 std. Significance vs RL (Single) under Welch's t with Holm-Bonferroni within each panel: * $p < 0.05$, ** $p < 0.01$. $N = 40$ per cell.

Figure 16: **Human study on Persuasion for Good.** Each subplot reports mean ± 1 std across $N=40$ Prolific participants per condition. * $p < 0.05$, ** $p < 0.01$ vs RL (Single) (Welch's t , Holm-Bonferroni within each panel; 2 pre-registered pairwise tests).

E Extended Experiments

E.1 Where Population Co-Training’s Gain Comes From

Population Co-Training was the top method on every benchmark in §4.2. Two design hypotheses follow from the theory: the buffer must be wide enough to preserve real cross-checkpoint disagreement (Q3a, pool size K), and the simulator reward must keep each checkpoint informative rather than re-collapsed on a new mode (Q3b, reward design). We probe each on τ^2 -bench Retail.

The buffer must be wide, but not stale. Q3a asks how many checkpoints the population should hold. Sweeping $K \in \{1, 3, 5, 10\}$ with one checkpoint every four training steps (Figure 17), $K=1$ reduces to a single moving target. Larger K mixes in older checkpoints whose simulator capability lags the current one; these stale partners dilute the gradient signal. $K=5$ and $K=10$ perform comparably at the top, with $K=5$ slightly ahead on both P4G and τ^2 -Retail; $K=1$ and $K=3$ fall meaningfully behind. The pool-size benefit plateaus around $K=5$: the pool’s value comes from preserving *current* disagreement, so stale checkpoints hurt at the same rate that fresh ones help.

The simulator reward must preserve variation. For Q3b we ablate the simulator’s reward design among three variants: *adversarial* ($r_\phi = -r_\pi$), *cooperative* ($r_\phi = r_\pi$), and a SPICE-style *curriculum* (r_ϕ peaks at a target within-batch variance). The two endpoints both collapse the simulator onto a new dominant mode and undo the moving target; only the curriculum reward preserves within-batch variation (Remark A.12) and yields gains over the no-co-training $K=3$ ensemble baseline. Full variants, numbers, and training curves are in Appendix E.8.

E.2 Reward-quadrant ablation

Our main-text results in Section 4 pair each task with the reward structure that best matches its role-and-objective configuration. Persuasion for Good uses the donation-based *adversarial* reward that the task definition already encodes. τ^2 -bench uses the *curriculum* reward studied in Appendix E.1, which shapes the simulator toward the within-batch variance regime of Remark A.12. CooperBench is symmetric: both sides share the binary task-success reward shipped with the benchmark, which is the

cooperative case. This appendix ablates the pairing. For each cell of the asymmetric/symmetric $\times$ adversarial/cooperative spectrum, we swap in a reward drawn from a different quadrant and rerun the same method set, to test whether the main-text conclusions depend on the default reward or on population training itself.

E.3 Empirical training dynamics across methods

Figure 19 extends Figure 3 with additional batch-level diagnostics. RL against a single frozen simulator collapses on every diagnostic the theory predicts. Zero-variance batches climb from 60% to over 85% (Lemma A.3). Policy entropy drops from 1.9 to 0.4 nats (Corollary 3.4). All-failure batches rise to 70% as the policy concentrates on a mode-exploit strategy that wins against no one. Within-simulator Verbalized Sampling slows the collapse but does not stop it. Cross-family ensembles ($K=3$) and the two Co-Training variants are the only methods that hold all four diagnostics healthy throughout training.

E.4 Per-benchmark training, eval, and entropy curves across all settings

We report the full training-time picture for every method on each benchmark, with the same three signals tracked in §3.3 (training reward, OOD eval reward, policy entropy). Three seeds per method; $\pm 1\sigma$ shading on the OOD panel. Each benchmark uses its own scale and method set.

P4G and τ^2 -bench Retail (Figures 20, 21) show the same qualitative pattern. RL (Single) takes the highest training reward against its training simulator and the lowest policy entropy, but its OOD eval slides back toward the untrained baseline. Verbalized Sampling and frozen ensembles trade a noisier, lower training reward for a higher steady OOD eval; Co-Training and Population Co-Training extend the OOD gain further and keep entropy in the 0.8–1.2 nat range throughout training.

CooperBench (Appendix Figures 23 and 24) adds a conversation-turn diagnostic. Cross-play against a fixed partner produces a turn-count curve that rises and then drops: once the policy finds a short strategy the frozen partner accepts, conversations get shorter. Self-play and Population Co-Training do not show this drop. The partner keeps adapting, so longer multi-turn solutions stay rewarded and the turn count keeps climbing. The 9B and 27B runs show the same qualitative pat-

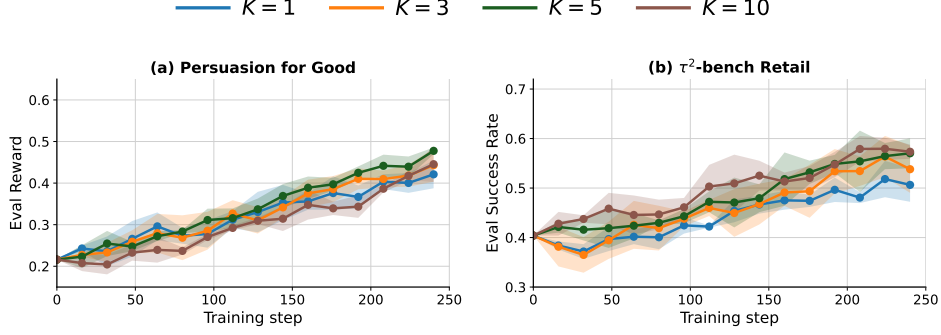


Figure 17: $K=5$ peaks above both extremes; the optimum is interior because stale checkpoints dilute the population. Eval reward over training for $K \in \{1, 3, 5, 10\}$ on P4G (left) and τ^2 -Retail (right); checkpoint cadence is every four training steps. Asymptote ordering: $K=5 > K=10 > K=3 > K=1$.

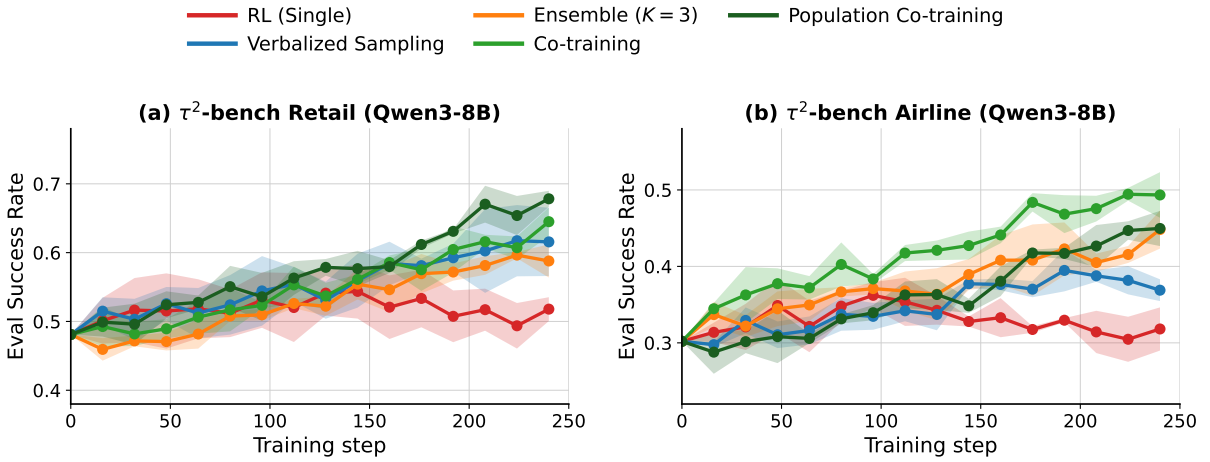


Figure 18: τ^2 -bench with Qwen3-8B. Scaling the trainable policy from Qwen3-4B-Instruct to Qwen3-8B preserves the ordering from Figure 4: RL (Single) collapses below the untrained baseline, every population-based method improves steadily, and Population Co-Training is the top curve. Noise reflects the smaller number of eval episodes in this setting.

tern, with the 9B curves visibly noisier reflecting the smaller model’s higher per-step variance on SWE-style tasks.

Collapse persists at the 8B scale. Scaling the trainable policy from Qwen3-4B-Instruct to Qwen3-8B does not eliminate simulator collapse: RL (Single) still saturates its training reward, peaks transiently on OOD, and crashes its policy entropy (Figure 22). The 8B run appears to collapse later than the 4B run, though we do not measure onset precisely; the qualitative pattern is the same.

E.5 Proof of Proposition 3.5

We prove Proposition 3.5 with the coupling argument of Theorem 3.2 and then bound tail-behavior coverage under γ -sharpening.

Proof of the gradient bound. Couple the trajectory τ_{VS} in M_{VS} (simulator samples from p_ϕ^{VS}) and

the trajectory τ_{ref} in M_{ref} (simulator samples from P) using shared agent randomness. At each simulator turn, given a matched prefix, sample (a_t^{VS}, a_t^{ref}) from the maximal coupling between $p_\phi^{VS}(\cdot | s_t, a_t^\pi)$ and $P(\cdot | s_t, a_t^\pi)$, which disagrees with probability at most $\eta(s_t, a_t^\pi)$ by hypothesis. By the union bound over simulator turns,

$$\Pr[\tau_{VS} \neq \tau_{ref}] \leq \mathbb{E}\left[\sum_{t=1}^H \eta(s_t, a_t^\pi)\right] = \bar{\eta}_H(\theta),$$

which gives $D_{TV}(P_{VS}^\theta, P_{ref}^\theta) \leq \bar{\eta}_H(\theta)$. The gradient bound then follows from the bounded-integrand TV inequality (Appendix A.2) applied to $f(\tau) = R(\tau)S_\theta(\tau)$ with $\|f\| \leq R_{\max}B$, as in the proof of Theorem 3.2. $\square$

Tail suppression under γ -sharpening. Together with γ -sharpening (Appendix A.3), the recovery assumption separates VS from direct prompting

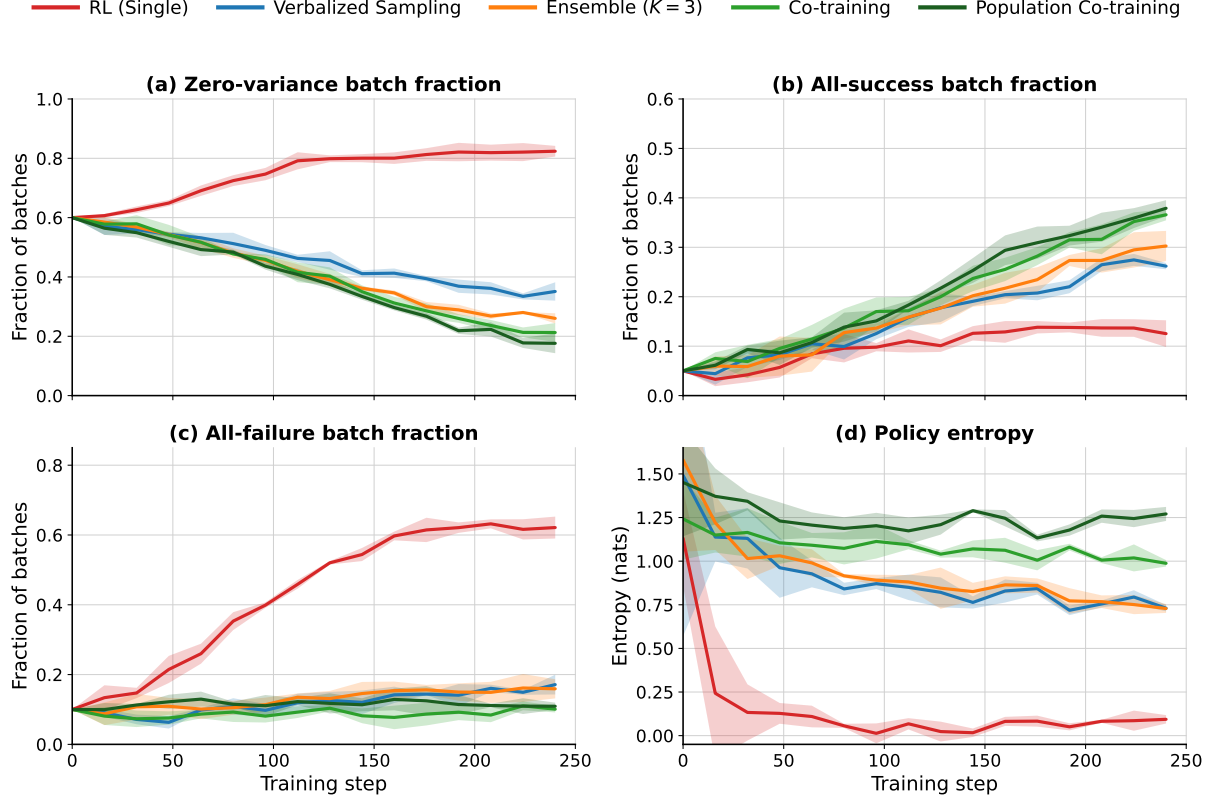


Figure 19: **Empirical training dynamics on τ^2 -bench Retail (full time series)**. Zero-variance batch fraction (top-left), all-success (top-right), all-failure (bottom-left), and policy entropy (bottom-right). Single-simulator RL (blue) blows up on every panel; Co-Training and Population Co-Training are the only methods that keep all four healthy. Shaded bands are $\pm 1\sigma$ over three seeds.

on tail behaviors. Direct prompting suppresses tail behaviors exponentially in γ , whereas VS preserves them up to η .

Proposition E.1 (γ -sharpening tail suppression). *Fix a state s and let $b^* = \arg \max_b P(b \mid s)$ with mass $m = P(b^* \mid s)$. Let $B \subset \mathcal{B}$ be a set of nonmodal behaviors with reference mass $P(B \mid s) = \rho$ and maximum per-behavior mass $\max_{b \in B} P(b \mid s) \leq \lambda m$ for some $\lambda \in [0, 1)$. Under direct γ -sharpened prompting $P_\gamma(b \mid s) \propto P(b \mid s)^\gamma$ with $\gamma > 1$ (Proposition A.2),*

$$P_\gamma(B \mid s) \leq \frac{\rho}{m} \lambda^{\gamma-1}.$$

Under VS with reference-recovery error η ,

$$p_\phi^{\text{VS}}(B \mid s) \geq \rho - \eta.$$

Proof. $\sum_{b \in B} P(b)^\gamma = \sum_{b \in B} P(b) \cdot P(b)^{\gamma-1} \leq \rho(\lambda m)^{\gamma-1}$ using $P(b) \leq \lambda m$ on B . The denominator of P_γ is at least the modal contribution m^γ , giving $P_\gamma(B) \leq \rho(\lambda m)^{\gamma-1}/m^\gamma = (\rho/m) \lambda^{\gamma-1}$. For the VS bound, $|p_\phi^{\text{VS}}(B) - P(B)| \leq D_{\text{TV}}(p_\phi^{\text{VS}}, P) \leq \eta$. $\square$

For the empirical range $\gamma \in [6, 66]$ (Appendix A.3) and λ bounded away from 1, $P_\gamma(B)$ is near zero while $p_\phi^{\text{VS}}(B)$ stays within η of ρ . By Proposition 3.5, the policy gradient under VS then approximates the reference-user gradient, with the gap controlled by $\bar{\eta}_H$.

Limits of the reference-recovery assumption. The reference-recovery assumption $D_{\text{TV}}(p_\phi^{\text{VS}}, P) \leq \eta$ is an empirical claim. It can fail if the simulator’s verbalized output systematically misses behavior types, for example if the K candidates always come from the cooperative half of the response distribution. The most direct test would sample the simulator under VS, cluster the responses into behavior types, and compare the empirical distribution to a behavior-level reference. We report transcript inspections in Appendix E.9 and leave quantitative behavior-coverage measurements to future work. Reference recovery is necessary but not sufficient, because P must also place mass on behaviors the policy needs to transfer. Proposition E.1 formalizes one such asymmetry. The real-user step requires

Persuasion for Good (Qwen3-4B-Instruct)

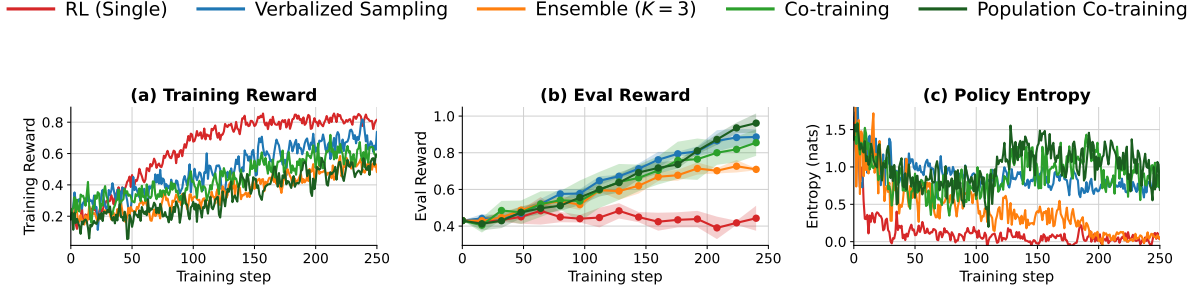


Figure 20: **Persuasion for Good: training, OOD eval, and policy entropy.** Six methods, three seeds each; $\pm 1\sigma$ shading on OOD. Training reward against the GPT-5-mini training simulator (a) is highest for RL (Single), which corresponds to the lowest agent entropy (c) and the worst OOD eval (b).

τ^2 -bench Retail (Qwen3-4B-Instruct)

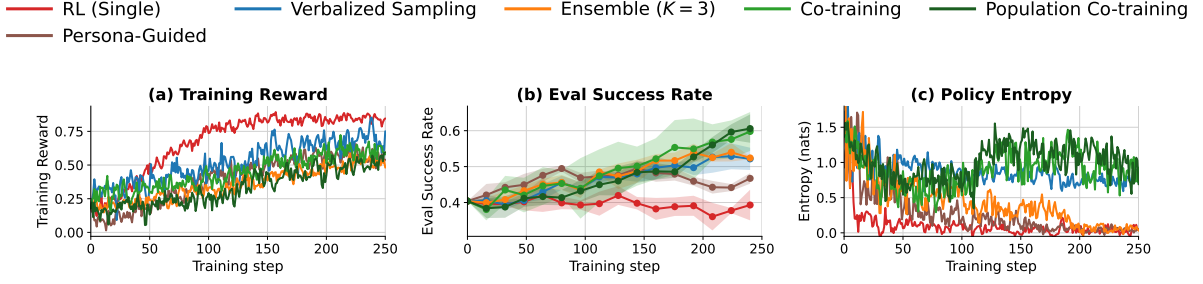


Figure 21: τ^2 -bench Retail: training, OOD eval, and policy entropy. Seven methods, three seeds each. Same qualitative pattern as Figure 20: RL (Single) wins training reward and loses OOD; Co-Training and Population Co-Training are best on the held-out panel.

an additional assumption $D_{TV}(P, P_{\text{real}}) \leq \kappa$, which VS itself does not establish and which the human study in Appendix D tests.

E.6 Verbalized Sampling mitigates simulator collapse

We isolate the effect of Verbalized Sampling (Zhang et al., 2025a) on the same training simulator used in our main experiments (GPT-5-mini). Figure 25 compares RL (Single) and RL (Single) + Verbalized Sampling under matched hyperparameters and three seeds each. Without VS, the policy follows the simulator-collapse signature established in §3.3: training reward climbs cleanly while OOD eval peaks and slides back, and policy entropy crashes to near zero. With VS the simulator is queried for a verbalized response distribution and the rollout is drawn from it, which restores enough simulator-side variance (Lemma A.3) to slow the geometric concentration of Corollary 3.4.

E.7 Verbalized Sampling on larger models

In this part we show training dynamics against the larger GPT-5 simulator. Given the limited budget constraint and fair comparison in Table 1, we use GPT-5-mini as the simulator across all settings in the main experiments. The original VS paper notes that smaller models suffer from “cognitive overload” when asked to verbalize a distribution while solving the task; against GPT-5 this is less of a concern, so VS’s mitigation effect is more pronounced.

Figure 26 shows the same three diagnostics against GPT-5. The collapse signature persists but is delayed: GPT-5 is less modal than GPT-5-mini, so the training reward climbs more slowly and tops out around 0.6–0.8, and the entropy collapse for RL (Single) sets in much later than in the GPT-5-mini setting. OOD eval for RL (Single) still peaks at ≈ 0.65 –0.67 and then degrades. With Verbalized Sampling the training reward is noisier and lower, but OOD climbs steadily and ends near 0.70, and entropy stays well above zero with substantial

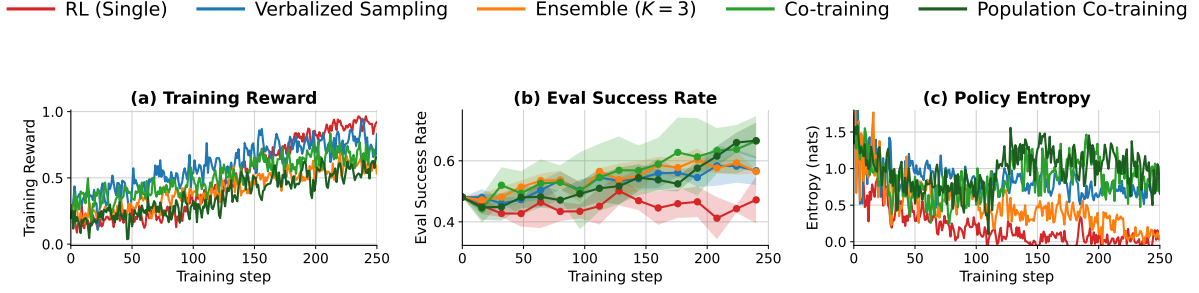
τ^2 -bench Retail (Qwen3-8B)

Figure 22: τ^2 -bench Retail (Qwen3-8B). Six methods, three seeds each; $\pm 1\sigma$ shading on OOD. The collapse pattern from Figure 21 appears at the 8B scale, qualitatively similar but visually later in training: RL (Single)’s training reward saturates, OOD eval peaks and slides, and policy entropy crashes. The per-step curves are also noisier across all three panels; the bigger policy has wider per-step variance at matched eval-episode counts.

variance throughout. The qualitative story matches the GPT-5-mini ablation in §E.6: VS reduces, but does not eliminate, simulator collapse.

E.8 Simulator-reward ablation

Q3b asks whether any choice of simulator reward yields the moving target, or whether the reward must be carefully shaped. We compare three simulator-reward variants (Table 9) against the no-co-training $K=3$ ensemble baseline (Figure 27). The two endpoints break the moving target, in mirrored ways. An *adversarial* reward ($r_\phi = -r_\pi$) collapses the simulator to $\sim 98\%$ refusal and drops eval reward to 0.07. A *cooperative* reward ($r_\phi = r_\pi$) pushes pushback to $\sim 2\%$, letting the policy reward-hack a trivial helper while eval reward drops from 0.27 to 0.17. Only the *curriculum* reward keeps opponent reward near 0.45, the regime of maximum within-batch variance (Remark A.12), and reaches ~ 0.40 eval. Both extremes collapse the simulator onto a new dominant mode, and once the mode is fixed again the simulator-collapse chain rebinds: the moving target stops moving. Population Co-Training helps *only when the simulator reward preserves variation across checkpoints*.

E.9 Within-batch rollouts: from diverse openings to a single strategy

We illustrate the policy-side signature of simulator collapse by inspecting three within-batch rollouts from the same starting context at three training stages (early / mid / late). Early in training the agent samples varied strategies and the simulator responds with varied content; by mid-training the

Table 9: Simulator-reward variants. r_π is the policy’s per-rollout reward; σ_π^2 its within-group variance. Curriculum follows SPICE-style variance shaping (Liu et al., 2025b).

Variant	Simulator reward r_ϕ
Adversarial	$-r_\pi$
Cooperative	r_π
Curriculum	$\exp\left(-\frac{(\sigma_\pi^2 - 0.25)^2}{0.02}\right)$

rollouts begin to share scaffolding; by late training all three rollouts within a batch are nearly word-for-word the same, which is what the geometric strategy-mass concentration of Corollary 3.4 looks like in transcript space.

E.10 Training on more models

We replicate the main P4G and τ^2 -bench Retail experiments with Olmo-3-7B-Instruct (Olmo et al., 2025) as the trainable agent, keeping every other element of the setup unchanged. Two differences relative to the Qwen3-4B-Instruct runs are notable. First, Olmo-3-7B-Instruct starts with higher base policy entropy than Qwen3-4B-Instruct (≈ 1.85 vs ≈ 1.55 nats); its RLHF profile is less sharpened. Second, its Base task performance is lower (0.34 vs 0.40 on τ^2 -Retail, 0.35 vs 0.43 on P4G), consistent with its smaller post-training budget. Despite these starting-point differences, the qualitative training dynamics match the Qwen3-4B-Instruct results: RL (Single) saturates training reward, peaks transiently on OOD eval, and crashes policy entropy from the higher initial point. Both Verbal-

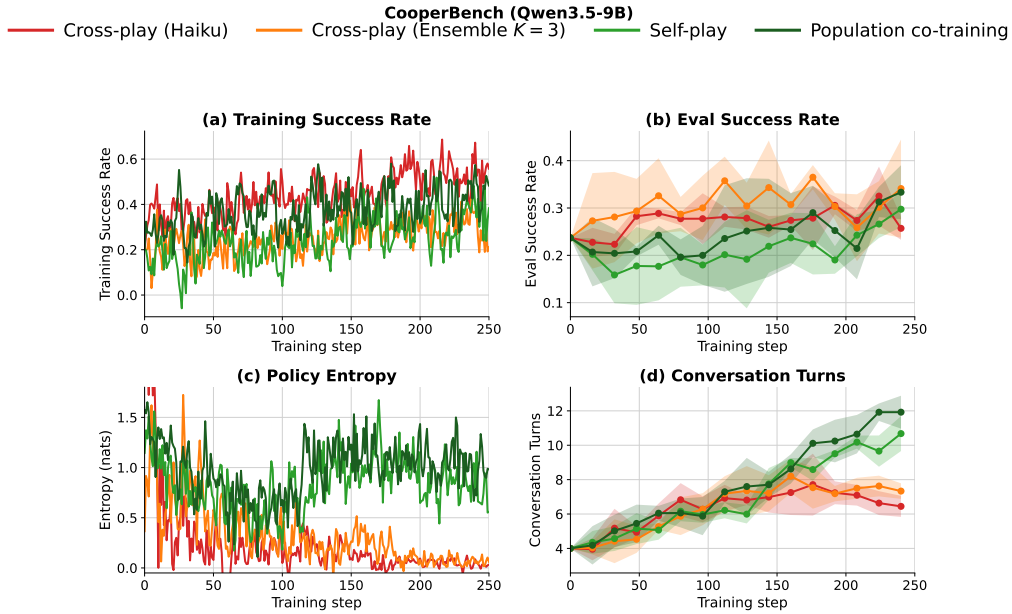


Figure 23: **CooperBench (Qwen3.5-9B): training, eval, entropy, and conversation turns.** The 9B run shows the same overfit signature as Figure 24 (Qwen3.5-27B) but with visibly more per-step variance, characteristic of SWE-style tasks at the smaller scale. Cross-play against a fixed Haiku partner or against a $K=3$ frozen ensemble plateaus and starts dropping in conversation turns as a short exploit strategy takes over; Self-play and Population Co-Training keep climbing.

ized Sampling and Co-Training recover most of the OOD gap and preserve entropy, and Population Co-Training is the strongest method on both benchmarks. Figures 28 and 29 report the full curves.

CooperBench (Qwen3.5-27B)

— Cross-play (Haiku) — Cross-play (Ensemble $K=3$) — Self-play — Population co-training

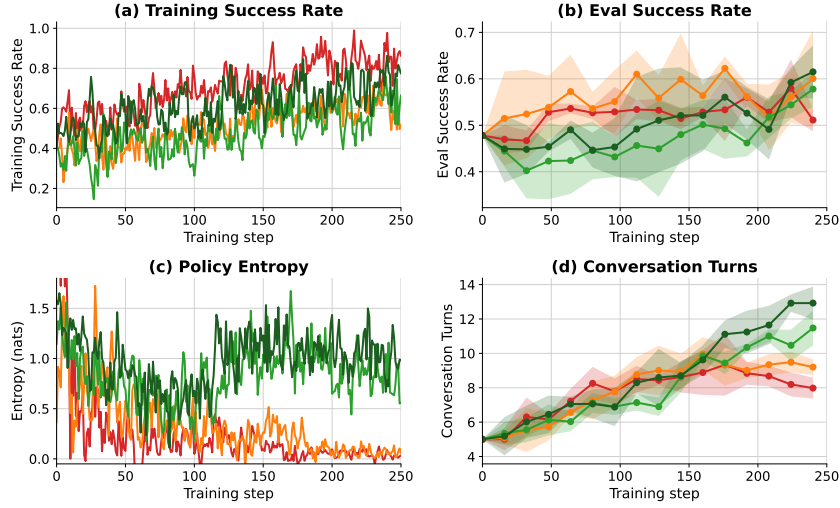


Figure 24: **CooperBench (Qwen3.5-27B): training, eval, entropy, and conversation turns.** Cross-play against a fixed Haiku partner or against a $K=3$ frozen ensemble shows the overfit signature on conversation turns: turns rise early as the policy learns to interact, then drop as a short exploit strategy takes over. Self-play and Population Co-Training avoid this regression; turns keep climbing as the partner co-evolves.

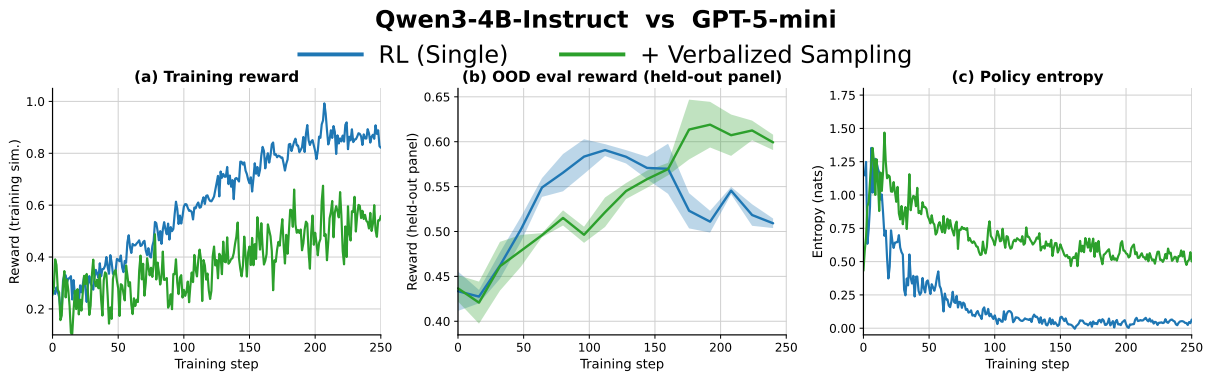


Figure 25: **Verbalized Sampling mitigates simulator collapse against GPT-5-mini.** Three seeds per setting. (a) Training reward: both runs climb, with +VS reaching a slightly higher plateau because the gradient signal is preserved on more batches. (b) OOD eval reward on the held-out 6-model panel ($\pm 1\sigma$ shaded): +VS peaks much higher and degrades far less than RL (Single), narrowing the gap to the untrained baseline. (c) Policy entropy: RL (Single) collapses to near zero, while +VS holds entropy near 0.7–0.8 nats throughout training. VS reduces but does not eliminate simulator collapse; the residual gap motivates the Co-Training experiments in §4.2.

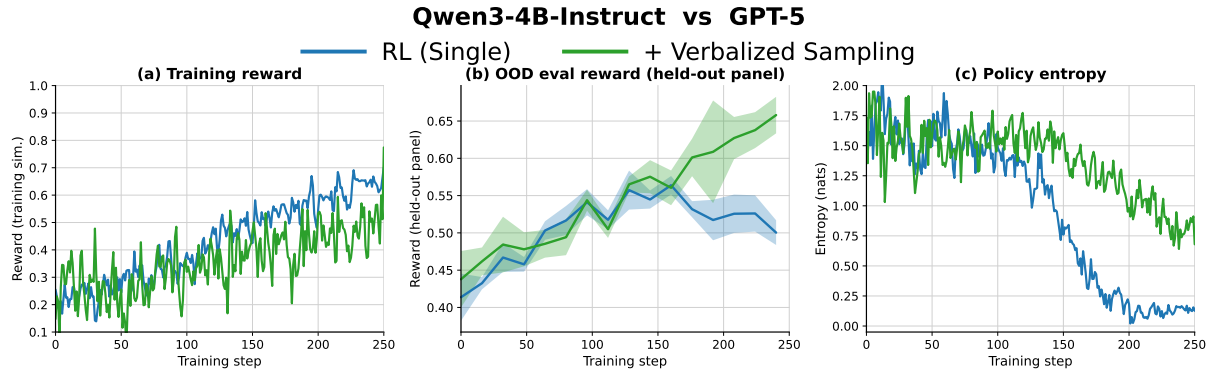


Figure 26: **Verbalized Sampling against GPT-5.** Three seeds per setting. **(a)** Training reward: RL (Single) climbs to a plateau in the 0.6–0.8 range; +VS is noisier and lower because the K-modal verbalized simulator gives less consistent reward signal. **(b)** OOD eval (held-out 6-model panel, $\pm 1\sigma$ shaded): RL (Single) peaks near 0.66 and degrades; +VS climbs more slowly but ends near 0.70. **(c)** Policy entropy: RL (Single) stays high for most of training and then crashes; +VS holds entropy above 1 nat throughout with substantial fluctuation.

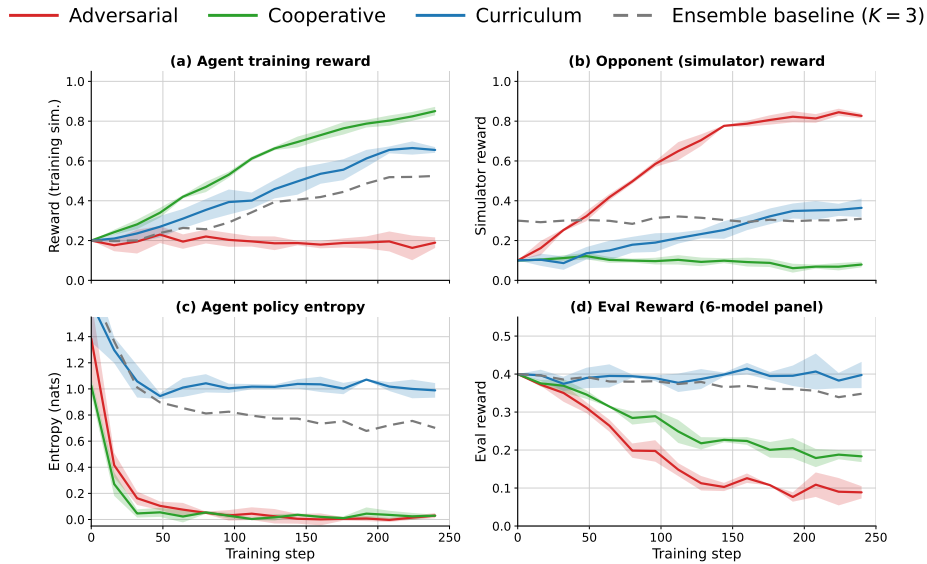


Figure 27: **Co-Training requires a carefully chosen simulator reward (τ^2 -bench Retail).** *Top-left:* agent training reward; cooperative reward (green) reaches the highest training reward by reward-hacking a trivially helpful simulator. *Top-right:* opponent reward; only the curriculum reward (blue) stays balanced near 0.45. *Bottom-left:* agent policy entropy; both extremes collapse to 0.01–0.04. *Bottom-right:* held-out eval reward; only the curriculum reward beats the no-co-training ensemble baseline.

Persuasion for Good (Olmo-3-7B-Instruct)

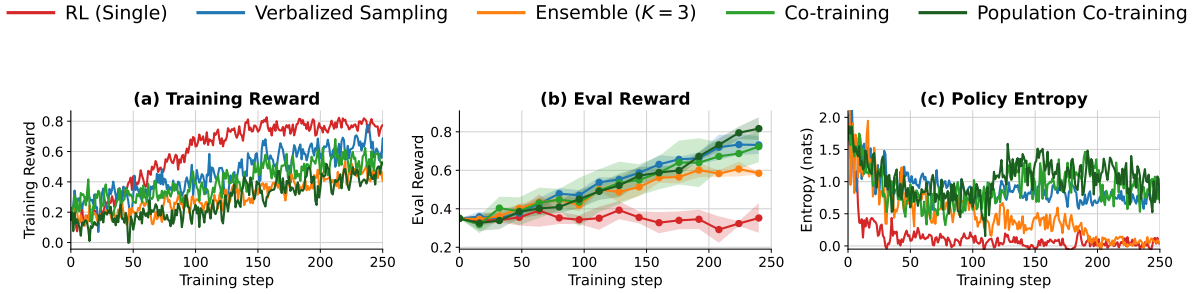


Figure 28: **Persuasion for Good (Olmo-3-7B-Instruct)**. Three panels: training reward, OOD eval reward, policy entropy. The qualitative simulator-collapse signature reproduces: RL (Single) saturates training reward, peaks transiently on OOD, and crashes its (higher initial) entropy; Verbalized Sampling, Ensemble, Co-Training, and Population Co-Training preserve entropy and close most of the held-out gap.

τ^2 -bench Retail (Olmo-3-7B-Instruct)

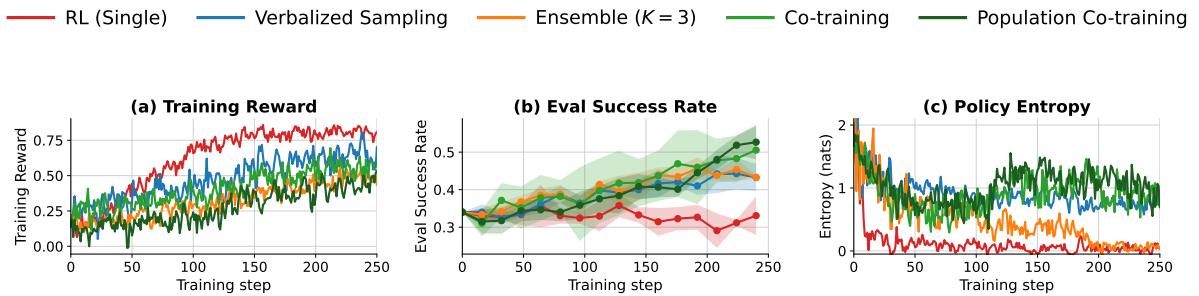


Figure 29: **τ^2 -bench Retail (Olmo-3-7B-Instruct)**. Same three panels as Figure 28. Olmo-3-7B-Instruct's Base success rate is lower than Qwen3-4B-Instruct's (0.34 vs 0.40), but the collapse-and-recovery pattern is the same.