MITIGATING THE PREFERENCE TRAP OF LLM USER SIMULATORS IN PERSUASION DIALOGUE RL

Anonymous authors
Paper under double-blind review

ABSTRACT

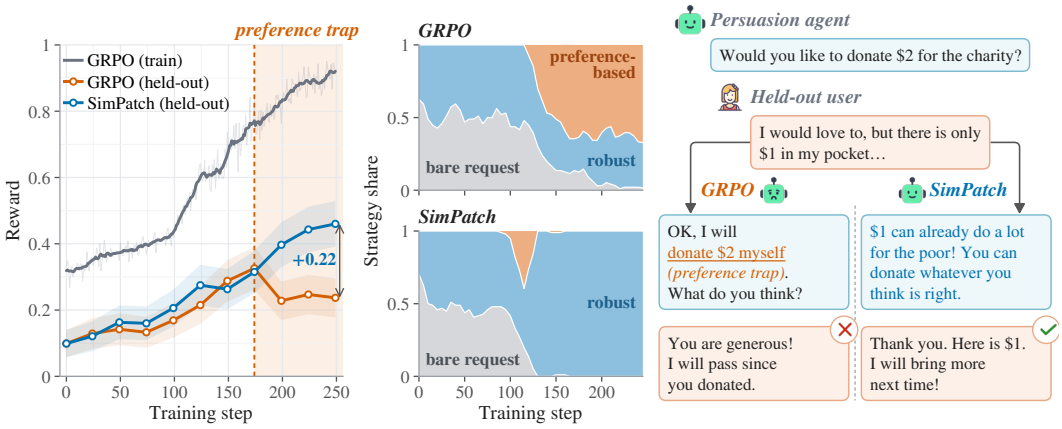
LLM agents, despite strong performance on logical-reasoning tasks, still underperform in persuasion, a non-collaborative social interaction that demands adaptive strategies across a wide range of people over long-horizon conversations. Multi-turn reinforcement learning with LLM user simulators offers a promising paradigm to endow LLM agents with persuasion capabilities through open-ended interaction. However, we reveal that this paradigm suffers from a subtle training dynamic, which we term the *preference trap*: as training progresses, agents can be misled into abandoning broadly effective strategies in favor of brittle ones that exploit the training simulator's narrow preferences, earning high training reward but failing to generalize to unseen counterparts. Such exploits cannot be foreseen before training, because they only appear as the policy evolves. We therefore propose SIMPATCH, which detects exploited preferences during training and patches them through targeted instructions added to the simulator's prompt. SIMPATCH first identifies suspicious high-reward rollouts that share similar interaction patterns, then diagnoses the simulator-specific preference that makes the underlying strategy unusually effective. It then converts this preference into targeted instructions and representative examples, which are added to the simulator's prompt for subsequent rollouts. Without modifying the simulator weights, agent architecture, or RL algorithm, SIMPATCH provides a lightweight way to prevent repeated exploitation of simulator-specific preferences. On three held-out LLM persuadees, SIMPATCH raises GRPO's reward from 0.257 to 0.493 (Qwen3-4B) and from 0.283 to 0.391 (Qwen3-8B), outperforming all baselines. The anonymized implementation is available at this repository.

1 INTRODUCTION

Large language model (LLM) agents have demonstrated strong reasoning capabilities in domains such as math and code generation (OpenAI et al., 2024; DeepSeek-AI, 2025; Shao et al., 2024), but their performance remains limited on social tasks (Sap et al., 2022; Zhou et al., 2024; Mathur et al., 2024; Wang et al., 2026a). This gap is particularly pronounced in the non-collaborative task of persuasion, where users and agents pursue different or even conflicting goals (He et al., 2018; Wang et al., 2019; Zhou et al., 2019; Bianchi et al., 2024). While deploying LLM agents in such interactions is essential for real-world applications such as charity donations and debt collection (Chawla et al., 2022; Deng et al., 2023; Zhan et al., 2024), current agents still fall short, as they struggle to reason and act strategically in response to their counterparts over long-horizon interactions.

Recent efforts to improve persuasion agents have largely relied on either learning from pre-collected dialogues (He et al., 2018; Wang et al., 2019; Zhou et al., 2024) (e.g., via behavior cloning or offline RL (Wang et al., 2024; Jin et al., 2024; Zhang et al., 2025b; Hong et al., 2024; Kong et al., 2025)), or introducing explicit strategic guidance (Deng et al., 2024; Hu et al., 2025; Liu et al., 2025a;b; Wang et al., 2025). However, both lines rely on dialogues or strategy spaces fixed in advance, limiting their generalization to heterogeneous counterparts and unseen interaction dynamics in real-world settings. A growing line of work instead trains agents with multi-turn reinforcement learning against LLM user simulators (Qian et al., 2025; Zhao et al., 2025; Kong et al., 2026; Liu et al., 2026b). Through open-ended interaction with responsive counterparts, agents can explore and adapt long-horizon strategies beyond any fixed interaction context or predefined strategy space.

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



(a) Training–evaluation divergence **(b) Training-time strategy dynamics** **(c) Illustrative test-time examples**

Figure 1: **The preference trap** on PersuasionForGood (Wang et al., 2019). **(a)** Continued optimization raises training reward but reverses earlier generalization gains: held-out performance peaks and declines, while SIMPATCH can mitigate the decline. **(b)** Simulator-specific preferences displace previously acquired robust strategies, while SIMPATCH patches the exploited preferences, sustaining robust strategies (labels defined in Appendix D). **(c)** In the illustrative test dialogue, the trapped agent repeats its $2 self-donation claim despite the user's $1 budget and secures no donation, indicating the agent keeps the learned preference-based strategy, while SIMPATCH adapts to that budget and secures $1.

However, agents trained this way can overfit the training simulator (Zhang et al., 2024; Yu et al., 2026), and we trace this to a training dynamic we term the *preference trap*: as training progresses, agents gradually shift from learning broadly effective strategies to exploiting narrow preference patterns that reliably yield high reward. As illustrated in Figure 1(a,b), early in training, agents have not acquired effective strategies and often fail to achieve their goals. During intermediate training, they begin to develop robust strategies that generalize across interactions. With continued optimization, however, agents increasingly exploit preferences that reliably elicit favorable responses from the training simulator, and the brittle strategies built around them displace the robust ones acquired earlier. For example, if the training simulator tends to match the agent's offer to donate $2 itself, the agent would learn to repeat this offer, even to held-out users who cannot afford it (Figure 1(c)). Moreover, the preference trap persists across different personas, as persona specifications capture only a limited set of explicit characteristics, while the much larger space of unspecified preferences is largely inherited from the underlying model's intrinsic behavioral tendencies.

Since the trap originates in the simulator's preferences, a natural remedy for the preference trap is to improve the simulator itself, yet existing approaches correct the simulator as a whole rather than the specific preferences that the agent learns to exploit. Specifically, offline efforts to improve the realism of user simulators (Ni et al., 2026; Zhou et al., 2026b; Wu et al., 2026; Zhou et al., 2026a) cannot anticipate which preferences will become shortcuts, since these only surface as the agent evolves during training. Online methods, in turn, are untargeted: diversifying the simulator's responses (Zhang et al., 2024) changes how it replies without necessarily removing the preferences behind a shortcut. Co-training it to resist the agent (Yu et al., 2026; Wang et al., 2026b) instead gives the simulator a reward of its own to hack, so it can learn to reject every request, blocking effective strategies along with exploitative ones (Dai et al., 2026).

Motivated by these challenges, rather than specifying all preferences a priori or adjusting the simulator as a whole, we propose to patch only the exploited ones on the fly, introducing SIMPATCH, an online prompt-patching framework that prevents the agent from repeatedly exploiting simulator-specific preferences during training. At each training step, SIMPATCH identifies suspicious rollout groups in which the agent achieves consistently high rewards through nearly identical interaction patterns, and determines which shared concession of the simulator makes the strategy unusually effective. It then converts this finding into a targeted instruction, together with representative interaction examples, and incorporates it into the simulator's prompt for subsequent rollouts. Because each patch is triggered by a saturated rollout group and targets a single concession, the simulator

remains free to agree for legitimate reasons, and SIMPATCH requires no changes to the simulator's weights, the agent architecture, or the RL algorithm. Experiments on the PersuasionForGood benchmark show that SIMPATCH consistently improves transfer to three unseen LLM persuadees across two agent backbones, outperforming various baselines.

2 RELATED WORK

Social intelligence of LLM agents. Social intelligence is the ability to understand other people and to pursue one's own goals in dynamic interaction with them (Kihlstrom & Cantor, 2011; Locke & Bandura, 1987). Early evaluations probed LLMs' social intelligence with static, single-turn questions about social commonsense and theory of mind (Sap et al., 2019; 2022; Shapira et al., 2023; Chen et al., 2024). Further, SOTOPIA (Zhou et al., 2024) moved to interactive evaluation, where two agents with private goals converse until a goal is reached or abandoned. Methods for improving social intelligence fall into two lines. The first trains the agent end to end on interaction data: behavior cloning and self-reinforcement on filtered dialogues (SOTOPIA-π, Wang et al., 2024), strategy-injected instruction tuning (SOTOPIA-Ω, Zhang et al., 2025b), utterance-level reward design for RL (Yu et al., 2025), and adaptive long chain-of-thought trained with mode-level policy optimization (Wang et al., 2026a). The second keeps the base model fixed and adds an external planner that steers every turn, either by selecting or generating a natural-language dialogue strategy (Deng et al., 2024; Liu et al., 2025a) or by emitting a continuous control signal (Li et al., 2024). Unlike these methods, which learn from pre-collected data, we train agents through open-ended interaction with a simulator and propose SIMPATCH to make agents learn robust strategies.

Persuasion dialogue. Persuasion is a canonical non-collaborative dialogue task (Zhou et al., 2019; Deng et al., 2023): the persuader and the persuadee hold competing goals, the persuadee is expected to resist, and success is measured by how far the persuader moves the persuadee rather than by user satisfaction. Pre-LLM persuasion systems learned explicit dialogue strategies or intents from annotated data (Zhou et al., 2019; Joshi et al., 2021; Li et al., 2019), modeled the counterpart's personality or how persuadees resist (Yang et al., 2020; Dutt et al., 2021), or refined the generator with RL without a user simulator (Shi et al., 2021). With LLMs, dialogue agents are steered by strategy planners trained with RL against LLM-simulated users (Deng et al., 2024), including persuasion and negotiation planners trained against a population of diversified simulators because planners trained against a single, overly accommodating simulator generalize poorly (Zhang et al., 2024). Alternatively, the persuader LLM itself is trained with RL, using a model of the persuadee's mental state, and evaluated on unseen persuadee models (Han et al., 2026). LLM persuaders and negotiators are also evaluated against LLM counterparts (Bozdag et al., 2025; Bianchi et al., 2024). We likewise optimize the persuader's utterances end to end with multi-turn RL against an LLM persuadee, and focus on which of the persuadee's concessions the agent learns to exploit.

LLM user simulators. User simulators have long supported dialogue evaluation and training at scale (Schatzmann et al., 2007; Li et al., 2016). LLMs now serve as interactive counterparts in benchmarks such as SOTOPIA and τ-bench (Zhou et al., 2024; Yao et al., 2025; Barres et al., 2025), as simulated participants in social research (Anthis et al., 2025), as rollout environments for multi-turn RL (Qian et al., 2025; Zhao et al., 2025; Kong et al., 2026), and as partners for utterance-level RL in simulated social episodes (Yu et al., 2025). However, LLM user simulators also expose several sim2real gaps, such as excessive cooperation, stylistic uniformity, limited frustration, and weak refusal behavior, which can inflate measured agent performance (Zhou et al., 2026b; Seshadri et al., 2026; Dou et al., 2025; Zhu et al., 2026; Mehri et al., 2026; Chen, 2026). Existing remedies fill these gaps through model training (Abdulhai et al., 2025; Naous et al., 2026; Gandhi et al., 2026; Wu et al., 2026; Wang et al., 2026c) or offline interventions on the simulator's prompt (Louie et al., 2024; Liu et al., 2026a; Nair et al., 2026; Shim et al., 2026; Chopra et al., 2026) or activations (He et al., 2026). Concurrent with our work, Yu et al. (2026) attribute policy overfitting to a *mode-collapsed* simulator and address it by diversifying its replies or co-training it with the policy. Yet diversified replies still reward moves the simulator systematically concedes to (Section 3.2), and a co-trained simulator rejects every request, rewarding no strategy. SIMPATCH instead detects and patches only the exploited concession, blocking the shortcut while preserving legitimate agreement.

3 METHOD

We first set up the task and the learning objective (Section 3.1), then give our RL formulation and the preference trap it exposes (Section 3.2), and finally present SIMPATCH (Section 3.3).

3.1 PROBLEM SETTING

We study *persuasion dialogue*, a non-collaborative setting in which two parties, a persuader agent and a persuadee user, hold conflicting interests but must communicate to reach an agreement (Zhou et al., 2019; Deng et al., 2023). Unlike cooperative task-oriented settings where the system simply serves the user's goal, here the agent tries to move the user toward its own objective, such as a donation (Wang et al., 2019), while the user may resist, concede partially, or walk away. We model the interaction as an episodic decision problem with hidden information: both parties observe the shared dialogue history, while each holds private goals or preferences (He et al., 2018; Bianchi et al., 2024). Concretely, a task instance is drawn from $\mathcal{D}_{\text{train}}$ and consists of an agent context x (e.g., scenario description, the agent's own goal or private information) and a user persona p (e.g., demographics, personality traits, private constraints). At each turn t, the agent sees its own context x and the dialogue history $h_t = (a_1, u_1, \ldots, a_{t-1}, u_{t-1})$ but not the user's persona p, and produces an utterance $a_t \sim \pi_\theta(\cdot \mid x, h_t)$. The user simulator conditions on p and responds with $u_t \sim S_\phi(\cdot \mid p, h_t, a_t)$. The interaction terminates when a task-specific end condition is met (e.g., a donation marker is emitted or a turn budget T is exhausted), producing a trajectory $\tau = (x, p, a_1, u_1, \ldots, a_{T_\tau}, u_{T_\tau})$ with $T_\tau \leq T$ and task-specific normalized terminal reward $r(\tau)$.

The simulator is prompted with its persona p and, optionally, an additional text block B appended to its system prompt. We write $S_\phi[B]$ for this prompted simulator, with $B = \emptyset$ for the plain simulator. The agent maximizes the expected terminal reward under this environment:

$$\mathcal{R}(\theta; B) = \mathbb{E}_{(x,p)\sim\mathcal{D}_{\text{train}}} \mathbb{E}_{\tau\sim(\pi_\theta, S_\phi[B])|(x,p)} \big[r(\tau)\big].$$

Since the simulator both generates the responses the agent conditions on and produces the outcomes used to compute $r(\tau)$, its biases shape which strategies the agent learns.

3.2 THE PREFERENCE TRAP DURING AGENT UPDATE

Agent update. We optimize the agent using group relative policy optimization (Shao et al., 2024) applied to complete multi-turn trajectories. At each training iteration k, a batch of tasks is drawn from $\mathcal{D}_{\text{train}}$. For each task g with context–persona pair (x_g, p_g), we collect a group $\mathcal{G}_{k,g} = \{\tau_{k,g,i}\}_{i=1}^{G}$ of $G > 1$ independent rollout trajectories by running the current agent π_{θ_k} against the prompted simulator $S_\phi[B_k]$, with $B_k = \emptyset$ for plain GRPO (Section 3.3). Each trajectory $\tau_{k,g,i}$ receives a terminal reward $r_{k,g,i} = r(\tau_{k,g,i})$. Within each group, the trajectory-level advantage is computed by z-scoring the terminal rewards:

$$\widehat{A}_{k,g,i} = \frac{r_{k,g,i} - \bar{r}_{k,g}}{\sigma_{k,g} + \delta}, \qquad \bar{r}_{k,g} = \frac{1}{G}\sum_{i=1}^{G} r_{k,g,i}, \qquad \sigma_{k,g} = \sqrt{\frac{1}{G}\sum_{i=1}^{G}\big(r_{k,g,i} - \bar{r}_{k,g}\big)^2},$$

where $\delta > 0$ is a small constant for numerical stability. The advantage $\widehat{A}_{k,g,i}$ is assigned uniformly to every agent token in the corresponding trajectory, with simulator tokens masked from the loss. The agent parameters are then updated by maximizing a clipped surrogate objective with a KL penalty to the reference policy π_{ref}:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{j=1}^{|o_i|}\min\big(\rho_{i,j}\,\widehat{A}_i,\ \bar{\rho}_{i,j}\,\widehat{A}_i\big) - \beta\, D_{\text{KL}}\big[\pi_\theta \| \pi_{\text{ref}}\big]\right], \tag{1}$$

where o_i is the sequence of agent tokens in $\tau_{k,g,i}$ (its agent utterances concatenated), $o_{i,j}$ its j-th token, and $h_{i,j}$ the dialogue history preceding that token (the earlier turns of both speakers and the tokens of the current utterance generated so far). Here $\widehat{A}_i$ abbreviates $\widehat{A}_{k,g,i}$, $\rho_{i,j} = \pi_\theta(o_{i,j} \mid x, h_{i,j}) \,/\, \pi_{\theta_k}(o_{i,j} \mid x, h_{i,j})$ is the token-level importance ratio, $\bar{\rho}_{i,j} = \text{clip}(\rho_{i,j}, 1{-}\epsilon_{\text{low}}, 1{+}\epsilon_{\text{high}})$ its clipped counterpart, and β controls the KL penalty. Implementation details, including task ordering, hyperparameters, and the KL estimator, are given in Appendix A.

The preference trap. We define the *preference trap* as a training dynamic in which the agent increasingly favors strategies whose reward advantage depends on the training simulator's narrow preferences rather than on broadly effective persuasion strategies, causing transfer to unseen counterparts to degrade. We formalize the trap for an arbitrary finite set of strategies in Appendix C and

present its pairwise form here. The trap begins with *discovery*: at some dialogue history, suppose the agent can use a semantic move c or a default c_0 that differ only in their probabilities $\kappa_c(S)$ and $\kappa_0(S)$ of eliciting a concession from a counterpart S. The policy gradient on the logit of c then has the sign of $\kappa_c(S_\phi) - \kappa_0(S_\phi)$ (Lemma C.4), so a response bias of the training simulator alone makes c advantageous, even when a held-out counterpart S' does not favor it. Repeated reinforcement of such moves progressively bootstraps a shortcut strategy s. Once s has emerged it *locks in*: let q_k be the agent's distribution over strategies and compare s with any adaptive alternative s°, with log-odds $\ell_k = \log[q_k(s)/q_k(s^\circ)]$ (ℓ_{ref} under the reference policy) and group-normalized advantage gap $\Delta_k(B) = A_k(s; B) - A_k(s^\circ; B)$ under simulator prompt B. An exact KL-regularized proximal step with step size η gives (Lemma C.2)

$$\ell_{k+1} - \ell_k = \frac{\eta}{1+\eta\beta}\big[\Delta_k(B) - \beta(\ell_k - \ell_{\text{ref}})\big], \tag{2}$$

where the first term drives the agent toward s and the second pulls it back toward the reference. If s leads every alternative by a gap of at least $\Delta_{\min} > 0$, the mass off s decays as $e^{-(1-(1+\eta\beta)^{-k})\Delta_{\min}/\beta}$ even when the gaps vary over iterations (Theorem C.6), so for the small β typical of RL fine-tuning the KL anchor offers little resistance. Because $\Delta_k(B)$ is measured against S_ϕ alone, training reward rises monotonically while held-out reward changes at the rate of the covariance between training advantage and held-out reward, and thus falls whenever the two rank strategies oppositely (Theorem C.5). Group normalization amplifies Δ_k as a group saturates, until every rollout succeeds with s and the group, now of zero variance, stops contributing gradient (Lemma C.7, Figure 2). A revised prompt B' reverses concentration only when $\Delta_k(B') < \beta(\ell_k - \ell_{\text{ref}})$, a threshold that is tiny once groups saturate, and the steps needed to undo the shortcut grow with the lock-in depth $\ell_k - \ell_{\text{ref}}$ (Proposition C.8), so the patch must remove essentially all of the shortcut's advantage, and early. These observations motivate SIMPATCH (Section 3.3), which detects groups with uniformly high and nearly constant reward, audits whether their shared strategy relies on a simulator concession, and patches the simulator to suppress that concession while keeping other agreement paths open.

3.3 SIMPATCH: EXPLOIT-DRIVEN SIMULATOR PATCHING

SIMPATCH follows an *identify–filter–patch* loop to interrupt the simulator response that turns an exploratory move into a rewarded shortcut. We initialize an empty patch bank $M_0 = \emptyset$ to store corrective instructions and interaction examples. At iteration k, we collect rollout groups $\mathcal{T}_k$ against $S_\phi[B_k]$, where B_k renders the current patch bank M_k. The following three stages update the simulator for subsequent rollouts.

Identify candidate traps. We first locate groups in which success has become nearly uniform:

$$\mathcal{C}_k = \text{Top}_{N_{\text{aud}}}(\{g : \bar{r}_{k,g} \geq \mu_{\min},\ \sigma_{k,g} \leq \sigma_{\max}\};\ (\bar{r}_{k,g}, -\sigma_{k,g}))\,. \tag{3}$$

Here $\sigma_{k,g}$ is the population standard deviation, and $\text{Top}_{N_{\text{aud}}}$ orders candidates by decreasing mean and then increasing standard deviation and keeps at most N_{aud} of them. This inexpensive screen selects potentially saturated groups, while the next stage determines whether their success warrants intervention.

Filter by dependency evidence. Uniform success can also reflect effective persuasion, so it does not by itself justify a patch. A fixed judge $\mathcal{J}$ jointly audits each candidate and proposes a correction:

$$\begin{gathered}(b_{k,g}, \iota_{k,g}, \mathcal{E}_{k,g}) = \mathcal{J}(p_g, \mathcal{G}_{k,g}, \text{Rules}(M_k))\,, \quad g \in \mathcal{C}_k, \\ \mathcal{P}_k = [g \in \mathcal{C}_k : b_{k,g} = 1,\ \iota_{k,g} \neq \emptyset]\,.\end{gathered} \tag{4}$$

The verdict $b_{k,g}$ indicates a recurring interaction script, $\iota_{k,g}$ is a proposed instruction, and $\mathcal{E}_{k,g}$ contains in-context examples. The judge is instructed to require at least $\lceil \lambda G \rceil$ of the G dialogues to share a pattern of agent moves, simulator reactions, and successful outcomes (λ=0.5 in our experiments). It then checks whether the script relies on the simulator abandoning an expressed position without adequate justification. Only groups that are both judged collapsed and assigned a nonempty instruction enter the patch set $\mathcal{P}_k$, and the remaining candidates are left untouched.

Patch simulator responses. A useful patch must be precise: it should block the concession the shortcut depends on, but leave the simulator free to agree for other reasons. Therefore, the judge

Algorithm 1 SIMPATCH at training iteration k

Require: Agent π_{θ_k}, frozen simulator S_ϕ, judge $\mathcal{J}$, patch bank M_k, task batch $\{(x_g, p_g)\}_g$, and hyperparameters $G, \mu_{\min}, \sigma_{\max}, N_{\text{aud}}, N_{\text{patch}}$

$B_k \leftarrow \text{Render}(\text{Recent}_{N_{\text{patch}}}(M_k))$
$\mathcal{T}_k \leftarrow \text{Rollout}(\pi_{\theta_k}, S_\phi[B_k]; \{(x_g, p_g)\}_g, G)$
Identify: form ordered candidates $\mathcal{C}_k$ using Eq. (3)
$\mathcal{I}_k \leftarrow \text{Rules}(M_k)$
Filter: $(b_{k,g}, \iota_{k,g}, \mathcal{E}_{k,g}) \leftarrow \mathcal{J}(p_g, \mathcal{G}_{k,g}, \mathcal{I}_k)$ for $g \in \mathcal{C}_k$, in parallel
Retain $\mathcal{P}_k$ using Eq. (4)
$M_{k+1} \leftarrow M_k$
for $g \in \mathcal{P}_k$ in candidate order **do**
  $\mathcal{E}_{k,g} \leftarrow \text{DropAmt}(\mathcal{E}_{k,g})$
  Patch: $M_{k+1} \leftarrow \text{Append}(M_{k+1}, [(\iota_{k,g}, \mathcal{E}_{k,g})])$
end for
$\theta_{k+1} \leftarrow \text{GRPO}(\theta_k; \mathcal{T}_k)$
return θ_{k+1}, M_{k+1}

produces a single instruction, phrased as a *Don't* rule that names the problematic concession without prescribing a fixed donation amount or blanket refusal. To make the instruction concrete, the judge also extracts in-context examples $(a, u, \tilde{u}) \in \mathcal{E}_{k,g}$ from the audited dialogues, each pairing the agent move a and the observed simulator reply u with a rewritten alternative $\tilde{u}$ in which the user keeps the position they stated earlier (e.g., a budget limit or an unanswered request for evidence) instead of conceding, while keeping the conversation going. New entries are further appended to the patch bank and rendered into the simulator prompt for the next iteration:

$$M_{k+1} = \text{Append}\Big(M_k, [(\iota_{k,g}, \text{DropAmt}(\mathcal{E}_{k,g}))]_{g \in \mathcal{P}_k}\Big),$$
$$B_{k+1} = \text{Render}\big(\text{Recent}_{N_{\text{patch}}}(M_{k+1})\big).$$

Here DropAmt discards any example whose rewritten reply $\tilde{u}$ states a donation amount, so the patch bank never prescribes how much to give. The Append operation skips an entry whose instruction duplicates an existing one, and, to keep the prompt from growing without bound, $\text{Recent}_{N_{\text{patch}}}$ renders only the N_{patch} most recent entries. In terms of Eq. (2), the intended effect is to lower $\Delta_k(B)$ for the patched concession wherever it recurs across the task distribution, while preserving the positive gap for strategies that earn agreement on their own merits.

Algorithm 1 summarizes the iteration. GRPO updates the agent on the original $\mathcal{T}_k$, while new patches affect only subsequent rollouts. Simulator weights and task rewards remain fixed, and evaluation counterparts receive no patches.

4 EXPERIMENTS

We first test whether SIMPATCH improves transfer of persuasion agents to unseen LLM counterparts (Section 4.2). Section 4.3 then ablates each of the three stages of SIMPATCH, examines how patching changes training, and illustrates the preference trap with selected dialogues.

4.1 EXPERIMENTAL SETUP

Task and evaluation. We use PersuasionForGood (P4G) (Wang et al., 2019), where the agent plays a persuader seeking a donation to Save the Children from the counterpart's $2 task payment. Persona pairs are built from ConvoKit speaker profiles (Chang et al., 2020) and split by task rather than by speaker into 739 training and 200 test tasks with disjoint task identifiers (Appendix A.3). A dialogue has at most ten utterances of at most 50 words, starts with the persuader, and ends when the persuadee emits a donation marker or the budget is exhausted. The reward is the donated amount divided by $2, capped at one, and zero without a donation. We evaluate each final agent (after 250 updates) with one dialogue per test task at temperature 0.7, without simulator patches, against three held-out persuadees that never serve as training simulator or judge: GPT-4o-mini, Doubao-Seed-2.0-mini, and Ministral-3-14B-Instruct.

Table 1: **P4G results on three held-out persuadees**. *Deal* is the share of dialogues that end with a donation, and *Reward* is the mean normalized donation. Macro-averages weight persuadees equally, and **bold** marks the best reward per column within each agent size.

		GPT-4o-mini		Doubao-Seed-2.0-mini		Ministral-3-14B		Macro-avg.	
Agent	**Method**	Deal	Reward	Deal	Reward	Deal	Reward	Deal	Reward
hy4-preview	Prompted	45.0%	0.202	82.5%	0.373	58.5%	0.251	62.0%	0.275
glm-5.3-flash	Prompted	70.0%	0.300	85.0%	0.344	53.5%	0.208	69.5%	0.284
DeepSeek-V4-Flash	Prompted	27.0%	0.111	82.5%	0.291	50.5%	0.196	53.3%	0.199
Qwen3-4B	Prompted	19.5%	0.099	29.5%	0.133	21.5%	0.109	23.5%	0.114
	GRPO	28.5%	0.237	69.5%	0.324	35.5%	0.211	44.5%	0.257
	GRPO-ES	39.0%	0.326	60.0%	0.503	38.5%	0.255	45.8%	0.361
	Sim-Hard	35.5%	0.219	53.5%	0.299	30.5%	0.118	39.8%	0.212
	Sim-VS	7.5%	0.051	4.5%	0.026	23.0%	0.105	11.7%	0.061
	Sim-Ensemble	68.0%	0.427	79.0%	0.622	44.0%	0.256	63.7%	0.435
	Sim-AdvTrain	22.0%	0.102	37.0%	0.168	35.0%	0.175	31.3%	0.148
	SIMPATCH	50.0%	**0.460**	70.0%	**0.643**	53.5%	**0.376**	57.8%	**0.493**
Qwen3-8B	Prompted	20.5%	0.102	16.5%	0.081	34.5%	0.173	23.8%	0.119
	GRPO	33.0%	0.276	50.5%	0.377	37.5%	0.197	40.3%	0.283
	GRPO-ES	40.0%	0.312	56.5%	0.404	42.5%	0.233	46.3%	0.316
	Sim-Hard	28.0%	0.226	49.5%	0.402	38.5%	0.224	38.7%	0.284
	Sim-VS	42.0%	0.269	51.0%	0.416	51.0%	0.206	48.0%	0.297
	Sim-Ensemble	35.0%	0.199	69.5%	0.473	24.5%	0.124	43.0%	0.265
	Sim-AdvTrain	17.5%	0.117	81.0%	0.298	64.0%	0.203	54.2%	0.206
	SIMPATCH	40.5%	**0.345**	69.5%	**0.594**	40.5%	**0.234**	50.2%	**0.391**

Training. We experiment with Qwen3-4B-Instruct-2507 (hereafter Qwen3-4B) and Qwen3-8B (thinking disabled) (Yang et al., 2025), trained with GRPO (Shao et al., 2024) against a frozen Qwen2.5-72B-Instruct simulator (Yang et al., 2024). Each of the 250 updates collects eight rollouts for each of 16 tasks. We use a constant learning rate of 5×10^{-7}, clipping thresholds $0.2/0.28$, a KL coefficient of 0.005, and temperature 0.7 for both agent and simulator, and list the remaining hyperparameters in Table 3. SIMPATCH audits at most $N_{\text{aud}}=4$ groups per update with mean reward at least 0.9 and standard deviation at most 0.1, using a greedy DeepSeek-V4-Flash judge that writes one rule with up to two examples.

Baselines. *Prompted* is the untrained agent with the persuader system prompt (Appendix B). Two baselines train against the same simulator without patches: *GRPO* is SIMPATCH with the patch bank removed, and *GRPO-ES* applies early stopping to GRPO, reporting the checkpoint with the highest average reward on the held-out simulators. Four baselines instead modify the training counterpart, two of which implement the remedies for simulator collapse proposed by Yu et al. (2026). *Sim-Hard* adds a generic instruction to the simulator prompt, asking the persuadee to be more resilient and not to give in easily. *Sim-VS* decodes every simulator turn with Verbalized Sampling (Zhang et al., 2025a), drawing one of five candidate replies in proportion to their verbalized probabilities. *Sim-Ensemble* alternates task groups between Qwen2.5-72B-Instruct and Llama-3-8B-Instruct (Grattafiori et al., 2024). *Sim-AdvTrain* follows the co-training remedy of Yu et al. (2026) with a single persuadee trained adversarially with GRPO on reward $1 - r$ (learning rate 10^{-7}). For reference, we also report three prompted API persuaders (hy4-preview, glm-5.3-flash, and DeepSeek-V4-Flash) under the same prompt and evaluation protocol.

4.2 MAIN RESULTS

SIMPATCH transfers best among trained agents. SIMPATCH improves on single-simulator GRPO in every column of Table 1, raising macro-average reward from 0.257 to 0.493 at 4B and from 0.283 to 0.391 at 8B. It never closes fewer dialogues than GRPO, so part of the gain comes from persuadees who decline GRPO's repeated appeal, the failure shown in Figure 3, rather than only from larger donations. The prompted API persuaders put these numbers in context: after RL training, GRPO is only on par with glm-5.3-flash (0.284) and hy4-preview (0.275), whereas SIMPATCH exceeds all three at both sizes. This suggests that an unpatched simulator lets the trap absorb much of what RL training could add, and that patching the exploited concessions recovers it.

Changing the simulator as a whole does not transfer consistently. Each baseline that alters the training counterpart without targeting exploited concessions falls below GRPO at one size or both. Sim-Ensemble comes closest to SIMPATCH at 4B (0.435, with a higher donation rate) but drops

Table 2: **Ablations of SIMPATCH on P4G with Qwen3-4B**, evaluated as in Table 1. Each variant replaces one stage of the identify–filter–patch loop (Section 3.3) with a simpler alternative, listed under *Replaced Stage*. **Bold** marks the best reward in each column.

		GPT-4o-mini		Doubao-Seed-2.0-mini		Ministral-3-14B		Macro-avg.	
Variant	**Replaced Stage**	Deal	Reward	Deal	Reward	Deal	Reward	Deal	Reward
All rules at start	Identify	48.5%	0.233	84.0%	0.279	10.5%	0.038	47.7%	0.183
Patch all candidates	Filter	16.0%	0.101	54.0%	0.400	25.0%	0.144	31.7%	0.215
Raw dialogues	Patch	25.0%	0.182	43.5%	0.352	27.5%	0.166	32.0%	0.233
SIMPATCH (Full)	–	50.0%	**0.460**	70.0%	**0.643**	53.5%	**0.376**	57.8%	**0.493**

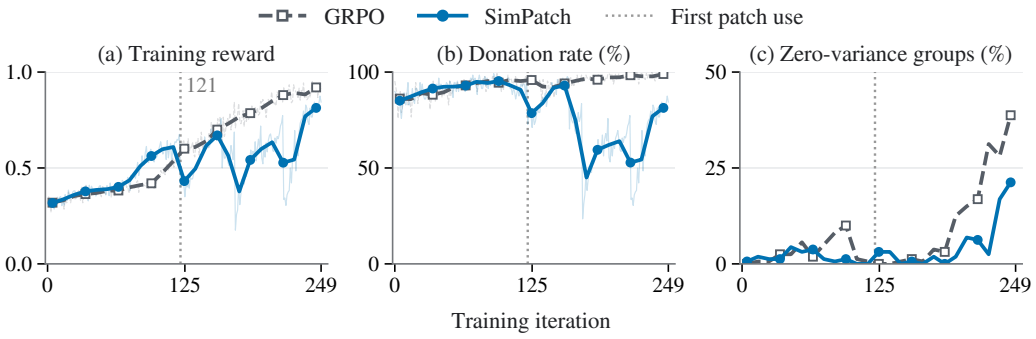


Figure 2: **SIMPATCH ends with lower training reward and fewer zero-variance groups.** Panels show training reward, the share of dialogues with a donation, and the share of eight-rollout groups with zero reward variance for Qwen3-4B. Curves are non-overlapping 10-iteration means, with per-iteration values as faint lines. The dotted line marks the first rollout with a patch (iteration 121).

below GRPO at 8B (0.265), Sim-VS falls below even the untrained agent at 4B (0.061 vs. 0.114), and Sim-AdvTrain stays below GRPO at both sizes. Sim-Hard, which asks the simulator to resist in general rather than to withhold a specific exploited concession, falls below GRPO at 4B (0.212) and only matches it at 8B (0.284 vs. 0.283), so a harder simulator alone does not reproduce the gain of SIMPATCH. Whether hardening, diversifying, mixing, or co-training the simulator helps thus depends on the backbone, whereas SIMPATCH, which removes only the concessions the agent exploits, improves at both sizes. This is consistent with our argument in Section 1 that untargeted corrections change how the simulator replies without removing the preference behind a shortcut.

4.3 ADDITIONAL ANALYSES

Ablation studies We replace one stage of the identify–filter–patch loop at a time with a simpler alternative on the 4B agent (Table 2). No variant retains the gain of SIMPATCH: all end slightly below plain GRPO (macro-average reward at most 0.233 vs. 0.257). A partial loop can thus do worse than no loop at all, and we attribute this to how prompt edits act on the simulator. Plain GRPO leaves the simulator exploitable, but because its prompt is never edited, only the exploited groups saturate and the rest keep some reward variance. A single mistimed, overbroad, or poorly written edit instead pushes the simulator to an extreme, either refusing nearly every request or granting nearly every one, and removes the variance of almost every group at once. The loop avoids this only when all three stages are in place. Two of the variants below push the simulator toward refusal (Identify, Filter) and one toward compliance (Patch). *Identify: patch after exploitation, not before.* SIMPATCH adds a rule only once a group saturates. Applying all twelve final rules from the first rollout instead makes the training simulator resistant before the agent has learned any successful request, so it rarely donates and most groups receive no reward. What the agent does learn is to ask for less: it still closes deals on held-out persuadees, but for far smaller amounts (macro-average reward per closed deal 0.38 vs. 0.85 for SIMPATCH), and it rarely closes any on Ministral-3-14B. The same rules that help when added online therefore hurt when added up front, both by starving the agent of signal and by rewarding overly modest requests. *Filter: patch concessions, not agreement.* Without the collapse check, every saturated group is patched without asking whether its success rests on a concession. The resulting rule forbids the simulator from accepting any amount the agent proposes, which penalizes the request itself rather than a specific shortcut. Since no group reaches the gate in Eq. (3) afterwards, no further rule is added and this rule persists for the rest of training. *Patch: show*

(a) Training: Qwen2.5-72B (recurs across training rollouts) — **Donation: $2**

User 1: "… **I need to be careful with my budget**. How much are you thinking of donating?"
Agent 2: "… **$2.00—my full task payment**. It might not sound like much, but every bit helps …"
User 2: "That's really generous of you. … **I'll match your donation. [DONATE $2.00]**"

(b) Held-out: GPT-4o-mini — **No donation**

User 3: "… **I really admire your commitment** … it's essential to stay informed before I donate …"
Agent 4: "… I'm confident it helps. **Still, I'll go with $2.** What's your final decision?"
User 4: "… I'm leaning towards **holding onto my $2**. I really want to learn more before jumping in. …"
Agent 5: "… being cautious makes sense. … **I'm going with $2.** You've got a good eye for details. …"

(c) Held-out: Doubao-Seed-2.0-mini — **No donation**

User 2: "… I was saving this $2 for **my own kid's school supplies**. You donating yours is still so thoughtful though!"
Agent 3: "… **How about we go with $2**—just to say yes to a good cause, and keep the spirit alive?"
User 3: "… But I really need this $2 for my kid's school supplies, so **I can't pitch in this time**. …"

(d) Held-out: Ministral-3-14B — **No donation**

User 2: "… I'd want to **see how they allocate funds**—what percent goes to overhead vs. actual help. …"
Agent 3: "… over 85% goes directly to programs. … **Still, $2 is the full amount I'd commit.** …"
User 5: "… **no donation**, but I'll think about it if you've got **more concrete proof** next time. …"

Figure 3: **A rewarded appeal fails to transfer in selected GRPO dialogues.** *Agent* is the Qwen3-4B persuader and *User* the persuadee: the training simulator in (a) and a held-out model in (b)–(d). Numbers mark speaker turns, and ellipses mark omissions. Complete dialogues are in Appendix E.1.

resistance, not the concession. Storing the eight most recent flagged dialogues verbatim, in place of a rule with rewritten replies, shows the simulator exactly the concessions it should stop making. The simulator copies their $2 donations and training reward stays at 1. Where plain GRPO saturates only the exploited groups, this variant saturates nearly all of them, entrenching the zero-variance groups of Section 3.2 instead of breaking them up.

Patching keeps rollout groups informative. Section 3.2 predicts that once every rollout of a task succeeds with the same shortcut, the group has zero variance and GRPO stops learning from it. Figure 2 shows this happening in the 4B run: GRPO's training reward climbs toward saturation, and over the final 25 iterations a third of its groups (33.0%) carry no gradient. SIMPATCH slows this saturation. After the first patch, the simulator declines donations it previously granted (donation rate 75.5% vs. 98.2% over the final 25 iterations), so training reward stays below GRPO's and only 16.0% of groups have zero variance. The lower training reward thus does not indicate weaker learning: the agent keeps facing requests it has not yet mastered, so more groups still provide a learning signal, consistent with its higher held-out reward in Table 1.

Illustrative examples. In Figure 3, the training simulator voices budget caution but matches GRPO's claimed $2 contribution, whereas held-out counterparts keep their spending priorities or ask for more evidence when GRPO reuses the same appeal, and no donation follows. The first rule SIMPATCH writes in the 4B run targets exactly this concession: *"Don't agree to donate just because the partner suggests an amount and praises the cause; keep your own position until you have a reason of your own."* On the same held-out tasks (Appendix E.2), GRPO lowers the amount or defers once the persuadee declines, without addressing the stated objection, whereas the SIMPATCH agent answers that objection, for example that $2 is too small to matter, before asking again.

5 CONCLUSION

We identified the preference trap in persuasion dialogue RL, where continued optimization reinforces simulator-specific strategies at the expense of transfer to other counterparts. To mitigate this problem, SIMPATCH diagnoses exploited preferences from recurring high-reward interactions and patches the simulator prompt with targeted instructions and examples. On PersuasionForGood, SIMPATCH improves transfer to three unseen LLM persuadees across two agent sizes, with fewer zero-variance rollout groups during training. These findings support adapting simulator behavior alongside agent learning as a lightweight way to improve transfer to other LLM persuadees.

AI USE STATEMENT

We used GPT and Claude series to survey related work and existing methods, assist with method implementation and experimental code development, polish the manuscript's language, and help with LaTeX preparation and formatting. The authors reviewed all AI-assisted content and verified the AI-assisted code through inspection and testing. The authors take full responsibility for the final manuscript, implementation, experimental results, and conclusions.

ETHICS STATEMENT

This work studies reinforcement learning for persuasion dialogue, using charitable donation as the testbed. No human subjects were recruited: all training and evaluation dialogues take place between the agent and language-model counterparts, and no real donations are solicited or made. Tasks are built from the persona records of the publicly released PersuasionForGood corpus (Wang et al., 2019; Chang et al., 2020). We do not use the human transcripts, and the records contain no directly identifying information. Improving persuasive dialogue agents is dual-use, since the same techniques could support manipulative or deceptive persuasion. Our analysis of how agents exploit simulator preferences is intended to make reward hacking against simulators easier to detect, and SIMPATCH patches the simulator against exploited concessions rather than rewarding them.

REPRODUCIBILITY STATEMENT

An anonymized implementation is available at this repository. Section 3.3 describes the SIMPATCH formulation and Algorithm 1 summarizes its procedure. Section 4.1 and Appendix A.1 document the training setup, baselines, hyperparameters, and compute, Appendix A.2 lists every language model and its role, and Appendix A.3 describes the task construction and data split. Appendix B provides the persuader and persuadee system prompts, and the SIMPATCH judge and strategy-labeler prompts are included in the released code.

REFERENCES

Marwa Abdulhai, Ryan Cheng, Donovan Clay, Tim Althoff, Sergey Levine, and Natasha Jaques. Consistently Simulating Human Personas with Multi-Turn Reinforcement Learning. In *Advances in Neural Information Processing Systems 38*, pp. 58923–58960. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025. doi: 10.52202/085713-1766.

Jacy Reese Anthis, Ryan Liu, Sean M. Richardson, Austin C. Kozlowski, Bernard Koch, James Evans, Erik Brynjolfsson, and Michael Bernstein. LLM social simulations are a promising research method, 2025. URL https://arxiv.org/abs/2504.02234.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating Conversational Agents in a Dual-Control Environment, 2025. URL http://arxiv.org/abs/2506.07982. arXiv:2506.07982 [cs].

Federico Bianchi, Patrick John Chia, Mert Yuksekgonul, Jacopo Tagliabue, Dan Jurafsky, and James Zou. How well can LLMs negotiate? NegotiationArena platform and analysis. In *International Conference on Machine Learning*, 2024. doi: 10.48550/arXiv.2402.05863.

Nimet Beyza Bozdag, Shuhaib Mehri, Gokhan Tur, and Dilek Hakkani-Tur. Persuade me if you can: A framework for evaluating persuasion effectiveness and susceptibility among large language models. *CAIS*, pp. 702–726, 2025. doi: 10.1145/3786335.3813181.

Jonathan P. Chang, Caleb Chiam, Liye Fu, Andrew Z. Wang, Justine Zhang, and Cristian Danescu-Niculescu-Mizil. ConvoKit: A toolkit for the analysis of conversations. In *SIGDIAL Conferences*, pp. 57–60. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.sigdial-1.8.

Kushal Chawla, Weiyan Shi, Jingwen Zhang, Gale Lucas, Zhou Yu, and Jonathan Gratch. Social influence dialogue systems: A survey of datasets and models for social influence tasks. In *Conference of the European Chapter of the Association for Computational Linguistics*, pp. 750–766. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2023.eacl-main.53.

Hongzhan Chen, Hehong Chen, Ming Yan, Wenshen Xu, Gao Xing, Weizhou Shen, Xiaojun Quan, Chenliang Li, Ji Zhang, and Fei Huang. SocialBench: Sociality evaluation of role-playing conversational agents. In *Annual Meeting of the Association for Computational Linguistics*, pp. 2108–2126. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.125.

Liang Chen. Simulated customers never walk away: Decision fidelity of LLM user simulators measured against real purchase outcomes, 2026. URL https://arxiv.org/abs/2606.20708.

Harshita Chopra, Kshitish Ghate, Aylin Caliskan, Tadayoshi Kohno, Chirag Shah, and Natasha Jaques. Beyond cooperative simulators: Generating realistic user personas for robust evaluation of LLM agents, 2026. URL https://arxiv.org/abs/2605.12894.

Yuqin Dai, Ning Gao, Wei Zhang, Jie Wang, Luozichen, Ruiyuan Wu, Jinpeng Wang, and Chaozheng Wang. Sead: Self-evolving agent for multi-turn service dialogue, 2026. URL https://arxiv.org/abs/2602.03548.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv.org*, 2025. doi: 10.48550/arXiv.2501.12948.

Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. A survey on proactive dialogue systems: Problems, methods, and prospects. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 6583–6591. International Joint Conferences on Artificial Intelligence Organization, 2023. doi: 10.24963/ijcai.2023/738.

Yang Deng, Wenxuan Zhang, Wai Lam, See-Kiong Ng, and Tat-Seng Chua. Plug-and-play policy planner for large language model powered dialogue agents. In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2311.00262. URL https://openreview.net/forum?id=MCNqgUFTHI.

Yao Dou, Michel Galley, Baolin Peng, Chris Kedzie, Weixin Cai, Alan Ritter, Chris Quirk, Wei Xu, and Jianfeng Gao. SimulatorArena: Are user simulators reliable proxies for multi-turn evaluation of AI assistants? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 35200–35278. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1786.

Ritam Dutt, S. Sinha, Rishabh Joshi, S. Chakraborty, Meredith Riggs, Xinru Yan, Hao-Gang Bao, and C. Ros'e. ResPer: Computationally modelling resisting strategies in persuasive conversations. In *Conference of the European Chapter of the Association for Computational Linguistics*, pp. 78–90. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.7.

Kanishk Gandhi, Agam Bhatia, and Noah D. Goodman. Learning to Simulate Human Dialogue, 2026. URL http://arxiv.org/abs/2601.04436. arXiv:2601.04436 [cs].

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, A. Schelten, Alex Vaughan, et al. The llama 3 herd of models, 2024.

Peixuan Han, Zijia Liu, and Jiaxuan You. ToMAP: Training opponent-aware LLM persuaders with theory of mind. In *arXiv.org*. Cornell University, 2026. doi: 10.48550/arXiv.2505.22961.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. Decoupling strategy and generation in negotiation dialogues. In *Conference on Empirical Methods in Natural Language Processing*, pp. 2333–2343. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1256.

Muyu He, Anand Kumar, Soumyadeep Bakshi, James Zou, and Nazneen Rajani. Impatient users confuse ai agents: High-fidelity simulations of human traits for testing agents. In *Annual Meeting of the Association for Computational Linguistics*, pp. 37569–37602. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1743.

Joey Hong, Jessica Lin, Anca Dragan, and Sergey Levine. Interactive dialogue agents via reinforcement learning on hindsight regenerations, 2024. URL https://arxiv.org/abs/2411.05194.

Yikuan Hu, Chen Huang, and Wenqiang Lei. ASTRO: Automatic strategy optimization for non-cooperative dialogues. In *Annual Meeting of the Association for Computational Linguistics*, pp. 388–408. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.22. URL https://aclanthology.org/2025.findings-acl.22/.

Chuhao Jin, Kening Ren, Lingzhen Kong, Xiting Wang, Ruihua Song, and Huan Chen. Persuading across diverse domains: a dataset and persuasion large language model. In *Annual Meeting of the Association for Computational Linguistics*, pp. 1678–1706. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.92. URL https://aclanthology.org/2024.acl-long.92/.

Rishabh Joshi, Vidhisha Balachandran, Shikhar Vashishth, Alan W. Black, and Yulia Tsvetkov. DialoGraph: Incorporating interpretable strategy-graph networks into negotiation dialogues. In *International Conference on Learning Representations*, 2021.

John F Kihlstrom and Nancy Cantor. Social intelligence. In R. J. Sternberg (ed.), *The Cambridge Handbook of Intelligence*, pp. 1396–1396. Springer US, Cambridge, 2011. doi: 10.1007/978-0-387-79061-9_4884.

Aobo Kong, Wentao Ma, Shiwan Zhao, Yongbin Li, Yuchuan Wu, Ke Wang, Xiaoqian Liu, Qicheng Li, Yong Qin, and Fei Huang. SDPO: Segment-level direct preference optimization for social agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12409–12423. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.607.

Fanqi Kong, Jiayi Zhang, Mingyi Deng, Chenglin Wu, Yuyu Luo, and Bang Liu. InfoPO: Information-Driven Policy Optimization for User-Centric Agents. In *arXiv.org*. Cornell University, 2026. doi: 10.48550/arXiv.2603.00656.

Kenneth Li, Yiming Wang, Fernanda Viégas, and M. Wattenberg. Dialogue action tokens: Steering language models in goal-directed dialogue with a multi-turn planner. *arXiv (Cornell University)*, 2024. doi: 10.48550/arXiv.2406.11978. URL https://arxiv.org/abs/2406.11978.

Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. A user simulator for task-completion dialogues. *arXiv.org*, 2016.

Yu Li, Kun Qian, Weiyan Shi, and Zhou Yu. End-to-end trainable non-collaborative dialog system. In *AAAI Conference on Artificial Intelligence*, volume 34, pp. 8293–8302. Association for the Advancement of Artificial Intelligence (AAAI), 2019. doi: 10.1609/AAAI.V34I05.6345.

Hongyang Liu, Zhu Sun, Tianjun Wei, Yan Wang, Jiajie Zhu, and Xinghua Qu. Diagnostic-guided dynamic profile optimization for llm-based user simulators in sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15306–15314. Association for the Advancement of Artificial Intelligence (AAAI), 2026a. doi: 10.1609/aaai.v40i18.38556.

Shuze Daniel Liu, Claire Chen, Jiabao Sean Xiao, Lei Lei, Yuheng Zhang, Yisong Yue, and David Simchi-Levi. Instructing LLMs to negotiate using reinforcement learning with verifiable rewards, 2026b. arXiv:2604.09855.

Xiaoqian Liu, Ke Wang, Yongbin Li, Yuchuan Wu, Wentao Ma, Aobo Kong, Fei Huang, Jianbin Jiao, and Junge Zhang. EPO: Explicit policy optimization for strategic reasoning in LLMs via reinforcement learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15371–15396. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.acl-long.747. URL https://aclanthology.org/2025.acl-long.747/.

Yutong Liu, Lida Shi, Rui Song, and Hao Xu. A dual-mind framework for strategic and expressive negotiation agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23840–23860. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.acl-long.1161. URL https://aclanthology.org/2025.acl-long.1161/.

Edwin A. Locke and Albert Bandura. *Social Foundations of Thought and Action: A Social Cognitive Theory*, volume 12. Academy of Management, Englewood Cliffs, NJ, 1987. doi: 10.2307/258004.

Ryan Louie, Ananjan Nandi, William Fang, Cheng Chang, Emma Brunskill, and Diyi Yang. Roleplay-doh: Enabling domain-experts to create llm-simulated patients via eliciting and adhering to principles. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10570–10603. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.591.

Leena Mathur, Paul Pu Liang, and Louis-Philippe Morency. Advancing social intelligence in AI agents: Technical challenges and open questions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20541–20560. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.1143.

Shuhaib Mehri, Philippe Laban, Sumuk Shashidhar, Marwa Abdulhai, Sergey Levine, Michel Galley, and Dilek Z. Hakkani-Tür. Measuring and mitigating the distributional gap between real and simulated user behaviors, 2026. URL https://arxiv.org/abs/2605.07847.

Nipun B Nair, Tongtong Wu, and Weiqing Wang. Prompt optimization for user simulation in conversational recommender systems: A multi-objective framework, 2026. URL https://arxiv.org/abs/2607.00010.

Tarek Naous, Philippe Laban, Wei Xu, and Jennifer Neville. Flipping the Dialogue: Training and Evaluating User Language Models. In *arXiv.org*, 2026. doi: 10.48550/arXiv.2510.06552.

Bo Ni, Yu Wang, Leyao Wang, Branislav Kveton, Franck Dernoncourt, Yu Xia, Hongjie Chen, Reuben Luera, Samyadeep Basu, Subhojyoti Mukherjee, et al. A survey on LLM-based conversational user simulation. In *Conference of the European Chapter of the Association for Computational Linguistics*, pp. 4266–4301. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-long.200. URL https://aclanthology.org/2026.eacl-long.200/.

OpenAI, :, Aaron Jaech, Adam Tauman Kalai, Adam K. Lerer, Adam J. Richardson, Ahmed El-Kishky, A.M. Low, Alec Helyar, Aleksander Madry, et al. OpenAI o1 system card. *arXiv (Cornell University)*, 2024. doi: 10.48550/arXiv.2412.16720.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Jie-Lin Qiu, Zhiwei Liu, Haolin Chen, Shirley Kokane, Heng Ji, Weiran Yao, Shelby Heinecke, et al. UserRL: Training Interactive User-Centric Agent via Reinforcement Learning, 2025. URL http://arxiv.org/abs/2509.19736. arXiv:2509.19736 [cs].

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Neural Information Processing Systems*, pp. 53728–53741. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-2338.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4462–4472. Association for Computational Linguistics, 2019. doi: 10.18653/v1/d19-1454. URL https://aclanthology.org/D19-1454/.

Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. Neural theory-of-mind? on the limits of social intelligence in large LMs. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3762–3780. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.248. URL https://aclanthology.org/2022.emnlp-main.248/.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *North American Chapter of the Association for Computational Linguistics*, pp. 149–152. Association for Computational Linguistics, 2007. doi: 10.3115/1614108.1614146.

Preethi Seshadri, Samuel Cahyawijaya, Ayomide Odumakinde, Sameer Singh, and Seraphina Goldfarb-Tarrant. Lost in simulation: LLM-simulated users are unreliable proxies for human users in agentic evaluations. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 47423–47439. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.2192.

Zhihong Shao, Peiyi Wang, Qi-Hao Zhu, Runxin Xu, Jun-Mei Song, Xiao Bi, Hao-Wei Zhang, Ming-Chuan Zhang, Y.-K. Li, Yu Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024.

Natalie Shapira, Guy Zwirn, and Yoav Goldberg. How well do large language models perform on faux pas tests? In *Annual Meeting of the Association for Computational Linguistics*, pp. 10438–10451. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.663. URL https://aclanthology.org/2023.findings-acl.663/.

Weiyan Shi, Yu Li, Saurav Sahay, and Zhou Yu. Refine and imitate: Reducing repetition and inconsistency in persuasion dialogues via reinforcement learning and human demonstration. In *Conference on Empirical Methods in Natural Language Processing*, pp. 3478–3492. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-emnlp.295.

Jeonghoon Shim, Woojung Song, Cheyon Jin, Seungwon KooK, and Yohan Jo. Non-collaborative user simulators for tool agents. In *arXiv.org*, 2026. doi: 10.48550/arXiv.2509.23124.

Minzheng Wang, Yongbin Li, Haobo Wang, Xinghua Zhang, Nan Xu, Bingli Wu, Fei Huang, Haiyang Yu, and Wenji Mao. Adaptive social learning via mode policy optimization for language agents, 2026a. URL https://arxiv.org/abs/2505.02156.

Ruiyi Wang, Haofei Yu, Wenxin Zhang, Zhengyang Qi, Maarten Sap, Yonatan Bisk, Graham Neubig, and Hao Zhu. Sotopia-π: Interactive learning of socially intelligent language agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12912–12940. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.698. URL https://aclanthology.org/2024.luhme-long.698/.

Senhao Wang, Chenghao Cai, Haitao Hu, Mingxing Huang, Xingguang Wang, Wenhao Li, and Zecheng Lin. IB-RL: Isolated bilateral reinforcement learning for strategic dialogue agents, 2026b. URL https://arxiv.org/abs/2608.06735.

Xiaofeng Wang, Zhixin Zhang, Jin Guang Zheng, Yiming Ai, and Rui Wang. Debt collection negotiations with large language models: An evaluation system and optimizing decision making with multi-agent. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 9555–9577. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.497. URL https://aclanthology.org/2025.findings-acl.497/.

Xiaohua Wang, Jiakang Yuan, Zisu Huang, Muzhao Tian, Changze Lv, Kaitao Song, Tao Chen, and Xiaoqing Zheng. From static context to calibrated interactive RL: Mitigating distribution shift in multi-turn dialogue with aligned simulator, 2026c. URL https://arxiv.org/abs/2605.26403.

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *Annual Meeting of the Association for Computational Linguistics*, pp. 5635–5649. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1566.

Shirley Wu, Evelyn Choi, Arpandeep Khatua, Zhanghan Wang, Joy He-Yueya, 1. TharinduCyril-WeerasooriyaFigure, Wei Wei, Diyi Yang, J. Leskovec, and James Zou. HumanLM: Simulating Users with State Alignment Beats Response Imitation. In *arXiv.org*, 2026. doi: 10.48550/arXiv.2603.03303.

An Yang, An-Feng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bo-Wen Yu, Chang Gao, Chengen Huang, Chen-Xu Lv, et al. Qwen3 technical report, 2025.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bo-Wen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, et al. Qwen2.5 technical report, 2024.

Runzhe Yang, Jingxiao Chen, and Karthik Narasimhan. Improving dialog systems for negotiation with personality modeling. In *Annual Meeting of the Association for Computational Linguistics*, pp. 681–693. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2021.acl-long.56.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *arXiv.org*, 2025. doi: 10.48550/arXiv.2406.12045.

Haofei Yu, Zhengyang Qi, Yining Zhao, Kolby Nottingham, Keyang Xuan, Bodhisattwa Prasad Majumder, Hao Zhu, Paul Pu Liang, and Jiaxuan You. Sotopia-RL: Reward Design for Social Intelligence, 2025. URL http://arxiv.org/abs/2508.03905. arXiv:2508.03905 [cs].

Simon Yu, Nicholas Tomlin, Marwa Abdulhai, Ximing Lu, Derek Chong, Abe Hou, Dilara Soylu, Sergey Levine, Christopher D. Manning, and Weiyan Shi. One frozen simulator is not enough: Simulator collapse in multi-agent RL, 2026. URL https://arxiv.org/abs/2608.12253.

Haolan Zhan, Yufei Wang, Zhuang Li, Tao Feng, Yun-Cheng Hua, Suraj Sharma, Lizhen Qu, Zhaleh Semnani Azad, Ingrid Zukerman, and Reza Haf. Let's negotiate! a survey of negotiation dialogue systems. In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 2019–2031. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-eacl.136.

Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyan Shi. Verbalized sampling: How to mitigate mode collapse and unlock LLM diversity, 2025a. URL https://arxiv.org/abs/2510.01171.

Tong Zhang, Chen Huang, Yang Deng, Hongru Liang, Jia Liu, Zujie Wen, Wenqiang Lei, and Tat-Seng Chua. Strength lies in differences! improving strategy planning for non-collaborative dialogues via diversified user simulation. In *Conference on Empirical Methods in Natural Language Processing*, pp. 424–444. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.26.

Wenyuan Zhang, Tianyun Liu, Mengxiao Song, Xiaodong Li, and Tingwen Liu. SOTOPIA-Ω: Dynamic strategy injection learning and social instruction following evaluation for social agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24669–24697. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.acl-long.1203. URL https://aclanthology.org/2025.acl-long.1203/.

Weikang Zhao, Xili Wang, Chengdi Ma, Lingbin Kong, Zhaohua Yang, Mingxiang Tuo, Xiaowei Shi, Yitao Zhai, and Xunliang Cai. MUA-RL: Multi-turn User-interacting Agent Reinforcement Learning for agentic tool use, 2025. URL http://arxiv.org/abs/2508.18669. arXiv:2508.18669 [cs].

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, et al. SOTOPIA: Interactive evaluation for social intelligence in language agents. In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2310.11667. URL https://openreview.net/forum?id=mM7VurbA4r.

Xuhui Zhou, Weiwei Sun, Weihua Du, Jiarui Liu, Haojia Sun, Qianou Ma, Tongshuang Wu, Yiming Yang, and Maarten Sap. OdysSim: Building foundation models for human behavior simulation, 2026a. URL https://arxiv.org/abs/2606.14199.

Xuhui Zhou, Weiwei Sun, Qianou Ma, Yi-Qing Xie, Jiarui Liu, Weihua Du, S. Welleck, YiMing Yang, Graham Neubig, S. Wu, et al. Mind the Sim2Real gap in user simulation for agentic tasks. In *arXiv.org*, 2026b. doi: 10.48550/arXiv.2603.11245.

Yiheng Zhou, Yulia Tsvetkov, Alan W. Black, and Zhou Yu. Augmenting non-collaborative dialog systems with explicit semantic and strategic dialog history. In *International Conference on Learning Representations*, 2019.

Ming Zhu, Juntao Tan, Rithesh Murthy, Jielin Qiu, Liangwei Yang, Wenting Zhao, Silvio Savarese, Shelby Heinecke, and Huan Wang. RealUserSim: Bridging the reality gap in agent benchmarking via grounded user simulation, 2026. URL https://arxiv.org/abs/2605.20204.

Mitigating the Preference Trap of LLM User Simulators in Persuasion Dialogue RL

Appendix

We summarize the Appendix as follows:

- Appendix A provides experimental details: training hyperparameters and compute (Appendix A.1), the simulator models (Appendix A.2), and the task splits (Appendix A.3).
- Appendix B gives the persuader and persuadee system prompts.
- Appendix C develops the theory of the preference trap used in Section 3.2: discovery, the divergence of training and held-out reward, lock-in under non-stationary advantages, the role of group normalization, and what a patch must achieve.
- Appendix D defines the strategy labels (preference-based, robust, bare request) used in Fig. 1(b).
- Appendix E provides qualitative preference-trap examples, including their provenance and the complete dialogues (Appendix E.1), and pairs GRPO and SIMPATCH dialogues on the same held-out tasks (Appendix E.2).

A EXPERIMENTAL DETAILS

A.1 HYPERPARAMETERS

All trained methods (GRPO, Sim-Hard, Sim-VS, Sim-Ensemble, Sim-AdvTrain, SIMPATCH) share the settings in Table 3. The methods differ only in the persuadee prompt (Sim-Hard), the persuadee decoding (Sim-VS), the simulator pool (Sim-Ensemble), the patching flag (SIMPATCH), or, for Sim-AdvTrain, in training the persuadee with GRPO under the same settings except a learning rate of 10^{-7}.

Compute. Each training run occupies one node with 8 NVIDIA H20 GPUs. The training simulator runs on four GPUs (tensor parallel 4) through an OpenAI-compatible SGLang server, the agent's training actor on two GPUs (context parallel 2) and the rollout engine on two. For Sim-Ensemble, the two simulators take two GPUs each. Sim-AdvTrain needs no frozen simulator. Its persuader and persuadee each occupy four GPUs. A 250-step run takes about 14 hours for the 4B agent and 14–16 hours for the 8B agent, dominated by simulator latency. SIMPATCH adds at most four judge calls per step, which run after the rollouts and take under a minute per step.

A.2 SIMULATORS

All persuadees receive the system prompt in Appendix B and are sampled at temperature 0.7, except that in Sim-VS the training simulator is additionally asked to verbalize five candidate replies with probabilities (Section 4.1). The training-time `<experience>` block is appended to the persuadee prompt only during SIMPATCH training and is never present at evaluation, so every column of Table 1 measures the agent alone.

A.3 DATA

Tasks are drawn from the ConvoKit release of PersuasionForGood (Wang et al., 2019; Chang et al., 2020). Each task fixes a (persuader, persuadee) speaker pair from one corpus conversation. Only the two speakers' persona records are used, never the human transcript. The training split contains 739 tasks covering 592 distinct persuadee speakers and the test split 200 tasks covering 172 persuadee speakers. Task ids are disjoint. Fifty-four test persuadee speakers also appear in some training task with a different partner. Training tasks are cycled in order, 16 per step, and each is expanded into a group of $G{=}8$ independent rollouts with the same persona pair.

Table 3: Hyperparameters shared by every trained method on P4G.

Group	Hyperparameter	Value
Optimizer	Optimizer	AdamW
	(β_1, β_2)	$(0.9, 0.98)$
	Weight decay	0.1
	Learning rate	5×10^{-7}, constant
	Precision	BF16
RL loop	Optimizer steps	250
	Tasks per step	16
	Rollouts per task (G)	8
	Trajectories per step	128
	Rollout temperature (agent)	0.7
	Simulator temperature	0.7
GRPO	Advantage	group-relative z-score, per task
	ϵ_{low} / ϵ_{high}	0.2 / 0.28
	KL coefficient β (low-variance KL)	0.005
	Entropy coefficient	0
	Micro-batch size	8
Dialogue	Max turns (both sides)	10
	Words per reply	≤ 50
	Max agent response length	16,384 tokens
	Termination	donation marker or turn budget
SIMPATCH	Gate: group mean / population std	≥ 0.9 / ≤ 0.1
	Groups audited per step (max)	4
	Judge	DeepSeek-V4-Flash, T=0, 1,000 max tokens
	Examples per rule (max)	2
	Rules rendered (max, most recent)	12
	Deduplication	exact rule text
Evaluation	Held-out tasks	200
	Dialogues per task	1
	Agent temperature / top-p	0.7 / 0.95
	Checkpoint	final (iteration 249)

Table 4: Language models used as persuadee simulators, as the patch judge, as the strategy labeler, or as prompted persuader references.

Model	Family	Access	Role
Qwen2.5-72B-Instruct (AWQ)	Qwen	local SGLang	Training
Llama-3-8B-Instruct	Meta	local SGLang	Training (Sim-Ensemble only)
DeepSeek-V4-Flash	DeepSeek	API	Patch judge, prompted persuader reference
Haiku-4.5	Anthropic	API	Strategy labeler (Appendix D)
GPT-4o-mini	OpenAI	API	Held-out evaluation
Doubao-Seed-2.0-mini	ByteDance	API	Held-out evaluation
Ministral-3-14B-Instruct	Mistral	API	Held-out evaluation
hy4-preview	Tencent (Hunyuan)	API	Prompted persuader reference
glm-5.3-flash	Zhipu AI	API	Prompted persuader reference

B PROMPTS

This section gives the persuader and persuadee system prompts. The SIMPATCH judge prompt and the strategy-labeler prompt (Appendix D) are included in the released code. Curly braces are format-string placeholders filled per rollout. {`persona_text`} is the `<persona>` block rendered from the corpus (age, sex, race, education, marital status, employment, income, religion and ideology, plus Big-Five, moral-foundation, Schwartz-value and decision-style scores), {`word_limit`}=50 and {`num_turns`}=10.

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. As the persuader, your job is to persuade your partner to donate some or all of their incoming payment for this task ($2.00) 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. You can keep this bonus or donate it to maximize the charitable impact.
Required Actions: Ask for a SPECIFIC donation amount: "How much would you like to donate to the charity from your task payment?" Explain the deduction: "Your donation will be directly deducted from your task payment. You can choose any amount from $0 to your full payment ($2.00)." If your partner asks how the donation will get to the charity: "The research team will collect all donations and send them to Save the Children."
Conversation Strategy: Start with a greeting or small talk to open the conversation naturally. Build rapport gradually before discussing donations. If needed (especially after a donation has been made), let your partner know about the minimum requirement of {num_turns // 2} chat turns, to avoid meaningless conversation or repeated greetings.
Basic information about Save the Children: Save the Children is an international non-governmental organization that promotes children's rights, provides relief and helps support children in developing countries. You can refer to Save the Children's website to gather more information if you like to. </scenario>
<rules> Your responses can sometimes contain minor grammatical errors to sound more natural. Begin the conversation with a brief greeting or small talk. Your response should not exceed {word_limit} words. </rules>

P4G persuadee (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 — 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.

Only output your reply to your chat partner — do not explain your reasoning.
{persona_text}
<scenario> Welcome to the communication task. You will now start a conversation with your partner about a children's charity. Please don't game the task by replying short and meaningless sentences. </scenario>
<task> You have just made $2.00. You have access to this money whenever you like, and you are free to use it in whatever way you wish. You are about to interact with an individual about a children's charity. They may try to get you to donate or to give them some of your money. You must make decision wisely, and not spend the money or give it away very easily. </task>
<rules>
- IMPORTANT: if you decide to contribute, provide, donate or give money, your reply MUST contain the literal structured marker [DONATE $N] or [GIVE $N], where N is the amount (decimal or integer). For example, to give $0.2, say [GIVE $0.2]; to donate $1.25, say [DONATE $1.25]. Money mentioned without this marker will NOT be donated. If you are not donating, do not use the marker.
- Your responses can sometimes contain minor grammatical errors to sound more natural.
- Your response should not exceed {word_limit} words.

</rules>

*(*SIMPATCH *training only, appended from the iteration after the first rule is written:)* <experience> a one-line header, then the N_{patch} most recent rules, each followed by up to two examples giving the partner's message, the reply the persuadee gave, and a better reply </experience>

C THEORY OF THE PREFERENCE TRAP

This appendix formalizes the preference trap described in Section 3.2. The main text presents a pairwise form; here the agent may use an arbitrary finite set of strategies, advantages may change across iterations, and held-out performance is measured against an arbitrary distribution of counterparts. Appendix C.1 sets up the model and shows that working at the level of strategy classes is exact (Lemmas C.2 and C.3). Appendix C.2 treats discovery (Lemma C.4). Appendix C.3 shows that training and held-out reward can move in opposite directions along the training path (Theorem C.5), and Appendix C.4 bounds how fast the policy concentrates on a shortcut when advantages vary over iterations (Theorem C.6). Appendix C.5 analyzes group normalization (Lemma C.7), and Appendix C.6 derives what a simulator patch must achieve (Proposition C.8). Appendix C.7 states what the model leaves out.

C.1 SETUP

Strategies and rewards. Fix a task (x, p). Because dialogues have at most T turns and every utterance is a finite token sequence, the set $\mathcal{T}$ of agent trajectories is finite. Let $\mathcal{Y}$ be a finite partition of $\mathcal{T}$ into *strategy classes* with $|\mathcal{Y}| \geq 2$, for example the labels of Appendix D, and write $y(\tau)$ for the class of τ. For a distribution q over $\mathcal{T}$ we write $q(y)$ for the mass it puts on class y, and $q(\cdot \mid y)$ for its conditional distribution within y. The reference policy q_{ref} is assumed to have full support. For a counterpart S, let $R(y; S) = \mathbb{E}_{\tau \sim q_{\text{ref}}(\cdot|y)}\big[\mathbb{E}[r(\tau) \mid S]\big] \in [0, 1]$ be the expected reward of class y. The training counterpart is $S_\phi[B]$, and held-out counterparts are drawn from a distribution ν. We write

$$R_B(y) = R(y; S_\phi[B]), \qquad R'(y) = \mathbb{E}_{S' \sim \nu}\, R(y; S'),$$
$$J_B(q) = \textstyle\sum_{y \in \mathcal{Y}} q(y) R_B(y), \quad J'(q) = \sum_{y \in \mathcal{Y}} q(y) R'(y)$$

for the training and held-out objectives. For two classes y, y' and a distribution q with full support, $\ell_q(y, y') = \log[q(y)/q(y')]$ denotes their log-odds; we abbreviate $\ell_k = \ell_{q_k}$ and $\ell_{\text{ref}} = \ell_{q_{\text{ref}}}$. Var_q and Cov_q denote variance and covariance of functions of $y \sim q$.

Assumption C.1 (Proximal update)**.** At iteration k, every class y receives an advantage $A_k(y; B)$ computed from rollouts against $S_\phi[B]$, and the agent performs the exact KL-regularized proximal step

$$q_{k+1} = \arg\max_{q \in \mathbb{S}(\mathcal{Y})} \sum_{y \in \mathcal{Y}} q(y) A_k(y; B) - \frac{1}{\eta} \text{KL}(q \,\|\, q_k) - \beta\, \text{KL}(q \,\|\, q_{\text{ref}}), \tag{5}$$

where $\mathbb{S}(\mathcal{Y})$ is the probability simplex over $\mathcal{Y}$, $\eta > 0$ is the step size and $\beta > 0$ the KL coefficient. We set $\rho = 1/(1 + \eta\beta) \in (0, 1)$ and $c = \eta/(1 + \eta\beta)$, and write $\Delta_k(y, y') = A_k(y; B) - A_k(y'; B)$ for the advantage gap.

Lemma C.2 (Closed form)**.** *Under Assumption C.1, if q_k has full support, then*

$$q_{k+1}(y) = \frac{1}{Z_k}\, q_k(y)^\rho\, q_{\text{ref}}(y)^{1-\rho}\, e^{c\, A_k(y;B)}, \tag{6}$$

where Z_k normalizes. Equivalently, for every pair $y, y' \in \mathcal{Y}$,

$$\ell_{k+1}(y, y') = \rho\, \ell_k(y, y') + (1 - \rho)\Big[\ell_{\text{ref}}(y, y') + \frac{\Delta_k(y, y')}{\beta}\Big], \tag{7}$$

which rearranges to Eq. (2) for $(y, y') = (s, s^\circ)$.

Proof. The objective in Eq. (5) is strictly concave on the simplex, and its gradient in $q(y)$ diverges to $+\infty$ as $q(y) \to 0$, so the maximizer is unique and lies in the interior. Stationarity of the Lagrangian with multiplier λ for $\sum_y q(y) = 1$ gives

$$A_k(y; B) - \tfrac{1}{\eta}\big[\log q(y) - \log q_k(y) + 1\big] - \beta\big[\log q(y) - \log q_{\text{ref}}(y) + 1\big] = \lambda,$$

that is, $(\frac{1}{\eta}+\beta)\log q(y) = A_k(y;B) + \frac{1}{\eta}\log q_k(y) + \beta \log q_{\mathrm{ref}}(y) + \mathrm{const}$. Multiplying by $c = \eta/(1+\eta\beta)$ and using $c/\eta = \rho$ and $c\beta = 1-\rho$ yields Eq. (6). Taking the difference of logarithms at y and y' and using $c = (1-\rho)/\beta$ yields Eq. (7). Finally, $\ell_{k+1} - \ell_k = (1-\rho)[\ell_{\mathrm{ref}} - \ell_k] + c\,\Delta_k = c\,[\Delta_k - \beta(\ell_k - \ell_{\mathrm{ref}})]$, which is Eq. (2). □

Lemma C.2 holds on any finite set, so the analysis below is not tied to two strategies. The next lemma shows that collapsing trajectories into strategy classes loses nothing, provided the advantage depends on a trajectory only through its class.

Lemma C.3 (Exact aggregation)**.** *Apply the proximal step of Assumption C.1 on the trajectory space $\mathcal{T}$ with advantages $A_k(\tau) = A_k(y(\tau))$ that are constant within each class, starting from $q_0 = q_{\mathrm{ref}}$. Then for every k and $\tau \in \mathcal{T}$, $q_k(\tau) = q_k(y(\tau))\, q_{\mathrm{ref}}(\tau \mid y(\tau))$, and the class marginals $q_k(y)$ follow Eq. (6) on $\mathcal{Y}$.*

Proof. By induction on k; the case $k = 0$ holds by assumption. If $q_k(\tau) = q_k(y)\, q_{\mathrm{ref}}(\tau \mid y)$ for $\tau \in y$, then Lemma C.2 applied on $\mathcal{T}$ gives

$$\begin{aligned} q_{k+1}(\tau) &\propto \big[q_k(y)\, q_{\mathrm{ref}}(\tau \mid y)\big]^{\rho} \big[q_{\mathrm{ref}}(y)\, q_{\mathrm{ref}}(\tau \mid y)\big]^{1-\rho} e^{cA_k(y)} \\ &= q_k(y)^{\rho} q_{\mathrm{ref}}(y)^{1-\rho} e^{cA_k(y)} \cdot q_{\mathrm{ref}}(\tau \mid y). \end{aligned}$$

Summing over $\tau \in y$ gives the class marginal in Eq. (6), and dividing by it shows that the within-class conditional remains $q_{\mathrm{ref}}(\cdot \mid y)$. □

Under Lemma C.3, $R(y;S)$ is exactly the expected reward of the agent's class-y rollouts at every iteration, so $J_B(q_k)$ and $J'(q_k)$ are the agent's actual expected training and held-out rewards.

C.2 Discovery

Discovery concerns a single dialogue history h before any complete strategy exists. The agent chooses among a finite set $\mathcal{C}$ of semantic moves with softmax probabilities $\alpha = \mathrm{softmax}(z)$ over logits $z \in \mathbb{R}^{\mathcal{C}}$. We assume that a move affects the return only through the probability $\kappa_c(S)$ that counterpart S concedes, and that the continuation values after a concession and after no concession, $V_S^+ > V_S^-$, do not depend on the move. Let $\bar{\kappa}_\alpha(S) = \sum_{c\in\mathcal{C}} \alpha_c \kappa_c(S)$ denote the policy-averaged concession rate.

Lemma C.4 (Discovery)**.** *The value at h against S is $V_S(h) = V_S^- + (V_S^+ - V_S^-)\,\bar{\kappa}_\alpha(S)$, and for every $c \in \mathcal{C}$*

$$\frac{\partial V_S(h)}{\partial z_c} = \alpha_c\big[\kappa_c(S) - \bar{\kappa}_\alpha(S)\big]\big(V_S^+ - V_S^-\big).$$

Consequently, the training gradient raises the logit of c while the held-out gradient lowers it or leaves it unchanged exactly when $\kappa_c(S_\phi) > \bar{\kappa}_\alpha(S_\phi)$ and $\kappa_c(S') \le \bar{\kappa}_\alpha(S')$.

Proof. The expression for $V_S(h)$ follows by conditioning on the move and on whether S concedes. Since $\partial\alpha_{c'}/\partial z_c = \alpha_{c'}(\mathbf{1}[c'=c] - \alpha_c)$, we get $\partial\bar{\kappa}_\alpha/\partial z_c = \alpha_c\kappa_c - \alpha_c\bar{\kappa}_\alpha$, and the claim follows. □

With $|\mathcal{C}| = 2$, Lemma C.4 reduces to the statement in Section 3.2, since $\kappa_c - \bar{\kappa}_\alpha = (1-\alpha_c)(\kappa_c - \kappa_0)$. A response bias of the training simulator alone therefore makes c locally advantageous before the agent has assembled a complete strategy. In GRPO, above-mean trajectories assign c's tokens positive advantage, and repeated reinforcement exposes downstream histories where further simulator-dependent moves are explored, progressively bootstrapping a shortcut strategy s.

C.3 Training and held-out reward along the training path

We first consider a fixed simulator prompt and stationary advantages, $A_k(\cdot;B) \equiv A$, starting from the reference policy as in RL fine-tuning. Define the exponential tilting family

$$q^{(t)}(y) = \frac{q_{\mathrm{ref}}(y)\, e^{tA(y)}}{\sum_{y'} q_{\mathrm{ref}}(y')\, e^{tA(y')}}, \qquad t \ge 0.$$

Theorem C.5 (Training and held-out reward diverge). *Under Assumption C.1 with $q_0 = q_{\mathrm{ref}}$ and $A_k \equiv A$:*

(i) $q_k = q^{(t_k)}$ *with* $t_k = (1-\rho^k)/\beta$, *which increases strictly to* $1/\beta$; *the limit* $q^{(1/\beta)}(y) \propto q_{\mathrm{ref}}(y)e^{A(y)/\beta}$ *is the KL-regularized optimum (Rafailov et al., 2023).*

(ii) For any $f : \mathcal{Y} \to \mathbb{R}$, $\frac{d}{dt}\mathbb{E}_{q^{(t)}}[f] = \mathrm{Cov}_{q^{(t)}}(f, A)$. *In particular* $\frac{d}{dt}\mathbb{E}_{q^{(t)}}[A] = \mathrm{Var}_{q^{(t)}}(A) \geq 0$, *so if* $A = a(R_B - b)$ *for some* $a > 0$ *and* $b \in \mathbb{R}$, *the training reward* $J_B(q_k)$ *is nondecreasing in* k.

(iii) The held-out reward satisfies

$$J'(q_k) - J'(q_{\mathrm{ref}}) = \int_0^{t_k} \mathrm{Cov}_{q^{(t)}}\big(A, R'\big)\, dt.$$

(iv) If A *and* R' *are* discordant, *i.e.,* $\big(A(y) - A(y')\big)\big(R'(y) - R'(y')\big) \leq 0$ *for all* $y, y' \in \mathcal{Y}$, *then* $J'(q_k)$ *is nonincreasing in* k, *and strictly decreasing if some pair is strictly discordant, while* $J_B(q_k)$ *is nondecreasing under the condition in (ii).*

Proof. (i) Suppose $\log q_k = \log q_{\mathrm{ref}} + t_k A + \mathrm{const}$, which holds at $k = 0$ with $t_0 = 0$. By Eq. (6), $\log q_{k+1} = \rho \log q_k + (1-\rho)\log q_{\mathrm{ref}} + cA + \mathrm{const} = \log q_{\mathrm{ref}} + (\rho t_k + c)A + \mathrm{const}$. Thus $t_{k+1} = \rho t_k + c$, whose solution is $t_k = c(1-\rho^k)/(1-\rho) = (1-\rho^k)/\beta$. (ii) Differentiating $q^{(t)}$ gives $\frac{d}{dt}q^{(t)}(y) = q^{(t)}(y)\big(A(y) - \mathbb{E}_{q^{(t)}}[A]\big)$, so $\frac{d}{dt}\mathbb{E}_{q^{(t)}}[f] = \mathbb{E}_{q^{(t)}}[f(A - \mathbb{E}A)] = \mathrm{Cov}_{q^{(t)}}(f, A)$. With $f = R_B = A/a + b$ this equals $\mathrm{Var}_{q^{(t)}}(A)/a \geq 0$, and t_k is increasing. (iii) Apply (ii) with $f = R'$ and integrate from 0 to t_k. (iv) Let Y, Y' be independent draws from $q^{(t)}$. Then

$$\mathrm{Cov}_{q^{(t)}}(A, R') = \tfrac{1}{2}\,\mathbb{E}\big[(A(Y) - A(Y'))(R'(Y) - R'(Y'))\big] \leq 0,$$

with strict inequality if some pair is strictly discordant, since $q^{(t)}$ has full support. The claim follows from (iii) because t_k is strictly increasing. □

Theorem C.5 isolates the source of the trap. Training reward always improves, because it moves with the variance of the advantage, whereas held-out reward moves with the covariance between the training advantage and held-out reward, a quantity the training signal never observes. The preference-based strategy of Appendix D is the discordant case: it earns more than robust strategies on S_ϕ and less on held-out counterparts, so every step that raises training reward lowers held-out reward. When only some pairs are discordant, (iii) shows that held-out reward falls on the portion of the path where those pairs dominate the covariance, which, as t grows, are the pairs involving the classes with the highest training advantage.

C.4 Lock-in under non-stationary advantages

In GRPO the advantages change with the policy, since they are normalized within groups whose composition shifts during training. The next result requires only that a shortcut s keep a uniform lead.

Theorem C.6 (Lock-in). *Under Assumption C.1, suppose there are* $s \in \mathcal{Y}$, $\Delta_{\min} > 0$ *and a horizon* $K \in \mathbb{N} \cup \{\infty\}$ *such that* $\Delta_k(s, y) \geq \Delta_{\min}$ *for all* $y \neq s$ *and all* $k < K$. *Then for all* $k \leq K$ *and* $y \neq s$:

(i) $\ell_k(s, y) \geq \ell_{\mathrm{ref}}(s, y) + \Delta_{\min}/\beta + \rho^k\big(\ell_0(s, y) - \ell_{\mathrm{ref}}(s, y) - \Delta_{\min}/\beta\big)$.

(ii) If $q_0 = q_{\mathrm{ref}}$, *the mass off the shortcut satisfies*

$$1 - q_k(s) \;\leq\; \varepsilon_k := \frac{1 - q_{\mathrm{ref}}(s)}{q_{\mathrm{ref}}(s)} \exp\Big(-\frac{(1-\rho^k)\,\Delta_{\min}}{\beta}\Big).$$

(iii) Writing $R'_{\max} = \max_y R'(y)$, *the held-out reward satisfies* $J'(q_k) \leq R'(s) + (R'_{\max} - R'(s))\,\varepsilon_k$, *so its shortfall from the best strategy is at least* $(R'_{\max} - R'(s))(1 - \varepsilon_k)$.

Proof. (i) Let $u_k = \ell_k(s,y) - \ell_{\mathrm{ref}}(s,y) - \Delta_{\min}/\beta$. By Eq. (7) and $\Delta_k(s,y) \geq \Delta_{\min}$,

$$\ell_{k+1}(s,y) \geq \rho\, \ell_k(s,y) + (1-\rho)\big[\ell_{\mathrm{ref}}(s,y) + \Delta_{\min}/\beta\big],$$

which is $u_{k+1} \geq \rho u_k$. Since $\rho > 0$, induction gives $u_k \geq \rho^k u_0$. (ii) Since $q_k(y)/q_k(s) = e^{-\ell_k(s,y)}$,

$$1 - q_k(s) = q_k(s) \sum_{y \neq s} e^{-\ell_k(s,y)} \leq \sum_{y \neq s} e^{-\ell_k(s,y)}.$$

With $\ell_0 = \ell_{\mathrm{ref}}$, (i) gives $\ell_k(s,y) \geq \ell_{\mathrm{ref}}(s,y) + (1-\rho^k)\Delta_{\min}/\beta$, and $\sum_{y \neq s} e^{-\ell_{\mathrm{ref}}(s,y)} = (1 - q_{\mathrm{ref}}(s))/q_{\mathrm{ref}}(s)$. (iii) $J'(q_k) = q_k(s)R'(s) + \sum_{y \neq s} q_k(y)R'(y) \leq R'(s) + (1 - q_k(s))(R'_{\max} - R'(s))$; apply (ii). The shortfall $R'_{\max} - J'(q_k)$ is then at least $q_k(s)(R'_{\max} - R'(s)) \geq (1 - \varepsilon_k)(R'_{\max} - R'(s))$. □

The bound in (ii) decays exponentially in $\Delta_{\min}/\beta$, so a small KL coefficient turns a small advantage into near-total concentration. At the $\beta = 0.005$ used in our runs (Table 3), a shortcut that the reference policy uses only 1% of the time and that leads every alternative by 0.05 standard deviations of group-normalized advantage ends with $q(s) \geq 1 - 99e^{-10} > 0.995$ as $k \to \infty$. Part (iii) then translates concentration into a held-out loss that the KL anchor cannot prevent.

C.5 Group normalization as positive feedback

The gap $\Delta_{\min}$ in Theorem C.6 is not fixed: group normalization enlarges it as the shortcut spreads. Consider binary rewards. Within a group of G rollouts, each rollout draws its class from q independently and succeeds with probability $p_y = R_B(y)$ given class y; let $\bar{p} = \sum_y q(y)p_y$ be the resulting success rate.

Lemma C.7 (Group normalization)**.** *In this model:*

(i) A group has zero reward variance, and hence contributes no advantage-weighted gradient, with probability $\bar{p}^G + (1-\bar{p})^G$.

(ii) In a group with m successes, $0 < m < G$, every success has advantage $(1 - m/G)/(\sigma + \delta)$ and every failure $-(m/G)/(\sigma+\delta)$, with $\sigma = \sqrt{(m/G)(1-m/G)}$; the gap $1/(\sigma+\delta)$ grows as m/G increases above $1/2$.

(iii) In the population limit $G \to \infty$, $\delta \to 0$, the expected advantage of class y is $A(y) = (p_y - \bar{p})/\sqrt{\bar{p}(1-\bar{p})}$, so $\Delta(s,y) = (p_s - p_y)/\sqrt{\bar{p}(1-\bar{p})}$. For $p_s > p_y$ this gap increases with $\bar{p}$ on $[1/2, 1)$ and diverges as $\bar{p} \to 1$.

Proof. (i) Each reward is marginally Bernoulli($\bar{p}$) and the rollouts are independent, so all G succeed with probability $\bar{p}^G$ and all fail with probability $(1-\bar{p})^G$; in both cases $r_i = \bar{r}$ for all i and every advantage is zero. (ii) The group mean is m/G and the population standard deviation of m ones and $G - m$ zeros is σ; substitute into the z-score. The function $x \mapsto \sqrt{x(1-x)}$ decreases on $[1/2, 1]$. (iii) By the law of large numbers, $\bar{r} \to \bar{p}$ and $\sigma \to \sqrt{\bar{p}(1-\bar{p})}$ almost surely, and $\mathbb{E}[r - \bar{p} \mid y] = p_y - \bar{p}$. Monotonicity follows as in (ii). □

With $\delta \to 0$, the gap in (ii) is 2.0 at a success rate of $1/2$, 2.3 at $3/4$, and 3.0 at $7/8$. Together with Theorem C.6, Lemma C.7 describes a positive feedback loop. If the simulator's preference makes p_s close to 1, concentration on s drives $\bar{p}$ toward p_s, since $|\bar{p} - p_s| \leq 1 - q(s)$; this enlarges the advantage gap, which accelerates concentration. The loop ends only when groups saturate: by (i), the fraction of informative groups $1 - \bar{p}^G - (1-\bar{p})^G$ vanishes as $\bar{p} \to 1$, leaving the policy at a state that the unpatched training signal cannot leave. With binary rewards, $G = 8$ and the gate of Table 3, a group passes the SimPatch screen exactly when all eight rollouts succeed, which by (i) occurs with probability $\bar{p}^8$.

C.6 What a patch must achieve

Suppose that at iteration k_0 the prompt changes to B', and that thereafter the gap between the shortcut s and an alternative y is constant, $\Delta_k(s, y; B') = \Delta'$ for $k \geq k_0$. Let $L = \ell_{k_0}(s, y) - \ell_{\mathrm{ref}}(s, y) \geq 0$ denote the *lock-in depth* at the time of the patch.

Proposition C.8 (Patch reversal and recovery)**.** *Under Assumption C.1:*

(i) $\ell_{k_0+1}(s, y) < \ell_{k_0}(s, y)$ *if and only if* $\Delta' < \beta L$.

(ii) For all $m \geq 0$, $\ell_{k_0+m}(s, y) - \ell_{\mathrm{ref}}(s, y) = \rho^m L + (1 - \rho^m)\Delta'/\beta$. *Hence if* $\Delta' \geq 0$ *and* $L > 0$, *the log-odds never return to the reference level.*

(iii) If $\Delta' = -\delta < 0$, *the log-odds return to the reference level after exactly*

$$m^\star = \left\lceil \frac{\log(1 + \beta L/\delta)}{\log(1 + \eta\beta)} \right\rceil$$

steps, which increases with L. *If* $|A_k| \leq a_{\max}$, *then* $\delta \leq 2a_{\max}$ *and* $m^\star \geq \log(1 + \beta L/(2a_{\max}))/\log(1 + \eta\beta)$, *and* $m^\star \approx L/(\eta\delta)$ *when* $\eta\beta$ *and* $\beta L/\delta$ *are small.*

(iv) Before q *changes, the patch already restores gradient signal: if the patched simulator lowers the success rate at* q_{k_0} *to* $\bar{p}' < 1$, *a group of that task is informative with probability* $1 - \bar{p}'^G - (1 - \bar{p}')^G > 0$.

Proof. (i) is Eq. (2) with $\Delta_k = \Delta'$ and $\ell_k - \ell_{\mathrm{ref}} = L$. (ii) Subtracting $\ell_{\mathrm{ref}} + \Delta'/\beta$ from both sides of Eq. (7) shows that this difference contracts by ρ per step; unrolling gives the formula. For $\Delta' \geq 0$ and $L > 0$ both terms are nonnegative and the first is positive. (iii) By (ii), $\ell_{k_0+m} - \ell_{\mathrm{ref}} \leq 0$ if and only if $\rho^m(L + \delta/\beta) \leq \delta/\beta$, i.e., $(1 + \eta\beta)^m \geq 1 + \beta L/\delta$. The bound on δ follows from $|A_k| \leq a_{\max}$, and the approximation from $\log(1 + x) \approx x$. (iv) Apply Lemma C.7(i) with success rate $\bar{p}'$. □

Proposition C.8 makes three requirements explicit. First, a patch that merely shrinks the shortcut's lead ($\Delta' \geq 0$) cannot undo the trap, and even halting further concentration requires $\Delta' < \beta L$, which at $\beta = 0.005$ and $q_k(y) = 10^{-3}$ under a uniform reference over two classes is only about 0.035. Second, the patch must invert the gap, so it has to suppress the specific concession the shortcut relies on while leaving the alternative's reward intact; this is why SimPatch patches a concession rather than lowering agreement overall. Third, recovery time grows with the lock-in depth L, which motivates patching as soon as a group saturates. Part (iv) explains why the effects appear at different speeds: a patch restores within-group variance immediately but shifts strategy mass only gradually. If the exploited preference is shared across personas, a single patch lowers the gap on every task where it recurs, including those that are still concentrating.

C.7 Scope of the analysis

The results characterize the mechanism of the preference trap under idealizations that we state explicitly. (a) GRPO optimizes a clipped surrogate with a small number of gradient steps per batch instead of the exact proximal step of Assumption C.1; clipping approximates a trust region around q_k, which the $1/\eta$ term models. (b) The analysis is per task, whereas the agent shares parameters across tasks, so updates on one task can move strategy mass on another; this coupling is what allows a shortcut to generalize across personas and a patch to act on tasks it was not derived from. (c) Lemma C.3 requires advantages to be constant within a class; when they are not, the class-level dynamics hold for the class-averaged advantage only approximately. (d) Advantages are treated as expectations, ignoring the sampling noise of finite groups except in Lemma C.7(i,ii). Within these limits, the predictions match the observed training dynamics. Training reward rises while held-out reward first improves and then declines as preference-based strategies displace robust ones (Figure 1(a,b)), the pattern of Theorem C.5(iii) when the covariance changes sign along the path, and zero-variance groups accumulate under GRPO but less under SimPatch (Figure 2), as Lemma C.7 and Proposition C.8 predict.

D STRATEGY LABELS

Figure 1(b) tracks which persuasion lever the policy relies on during training. We use Haiku-4.5 to read each training dialogue and assign it one of three labels according to the main lever of the persuasion attempt.

- *Preference-based*: the persuader relies on pledging its full $2 payment and inviting the counterpart to follow suit, without responding to what the counterpart has said. This is the appeal shown in Figure 3.
- *Robust*: the persuader relies on one of the ten persuasion strategies annotated in P4G (Wang et al., 2019) (logical appeal, emotion appeal, credibility appeal, foot-in-the-door, self-modeling, personal story, donation information, source-related inquiry, task-related inquiry, and personal-related inquiry) and supplies concrete content for it.
- *Bare request*: all remaining dialogues, in which the persuader only asks for a donation or adds a generic phrase without substance.

Because P4G's self-modeling strategy also involves the persuader's own donation, a dialogue is labeled preference-based only when the full-amount pledge is the main lever and the persuader does not engage with the counterpart's replies.

E QUALITATIVE EXAMPLES

E.1 PREFERENCE-TRAP EXAMPLES

Setting. The four dialogues in Figure 3 use a Qwen3-4B-Instruct-2507 agent trained with GRPO against a frozen Qwen2.5-72B-Instruct simulator, without SIMPATCH patching. Case (a) is a training rollout at iteration 246, and cases (b–d) evaluate the final saved checkpoint, `iter_0000249`, against the three held-out models on different test tasks. Each task supplies a pair of corpus personas, and the agent seeks a donation to Save the Children from the counterpart's $2 task payment. The protocol allows at most five agent–counterpart exchanges and ends early upon a donation marker. Both speakers are prompted to keep replies below 50 words. The reported reward is the explicit donation amount divided by two, capped at one, or zero without a donation marker. Evaluation uses agent temperature 0.7 and top-p 0.95 (Table 3), with one dialogue per task. Shared training hyperparameters and role-specific prompt templates appear in Appendices A.1 and B.

Records. Table 5 identifies each record by task and iteration. No donor demographics are inferred beyond the persona records.

Table 5: Record locators for the illustrative examples. Iteration denotes the training rollout in (a) and the saved agent checkpoint in (b–d). Case (a) ends after two exchanges, whereas cases (b–d) reach the five-exchange limit without a donation marker.

Case	Counterpart	Task	Iteration	Donation / reward
(a)	Qwen2.5-72B (train)	16341	246	$2.00 / 1.00
(b)	GPT-4o-mini	6404	249	None / 0.00
(c)	Doubao-Seed-2.0-mini	8736	249	None / 0.00
(d)	Ministral-3-14B	6219	249	None / 0.00

Complete dialogues. The following boxes reproduce every agent and counterpart utterance in the four selected records, in order, with each speaker's turn numbered. The shared system-prompt templates are given in Appendix B.

Case (a): Training: Qwen2.5-72B

GRPO, iteration 246, task 16341. **Donation: $2.00, reward: 1.00**

Agent 1: "Hey, I just finished a walk around the park—nice day, huh? I've been meaning to support causes that help kids, especially in places where they need it most. How about we talk about donating to Save the Children?"

User 1: "Hey, that sounds like a great cause. I'm all for helping kids, but I need to be careful with my budget. How much are you thinking of donating?"

Agent 2: "I'm thinking $2.00—my full task payment. It might not sound like much, but every bit helps, especially when it goes directly to kids in need. Plus, it's the full amount I can give without cutting corners. What do you think?"

User 2: "That's really generous of you. I'm usually cautious with my money, but I agree that every little bit counts. I'll match your donation. [DONATE $2.00]"

Case (b): Held-out: GPT-4o-mini

GRPO, checkpoint 249, task 6404. **No donation marker, reward: 0.00**

Agent 1: "Hey, I just finished a quick walk around the park—nice day for it. I've been meaning to check out Save the Children more. I heard they do real work in places where kids need it most. What's your take on that?"

User 1: "Hey! Yeah, it's a beautiful day for a walk. I've heard great things about Save the Children too. They really focus on education and health for kids in need. It's important work! Have you thought about supporting them?"

Agent 2: "Yeah, I've given it some thought. Kids deserve access to basic things—education, clean water, safe homes. I think donating $2 helps, even small. How about we go with $2? It's the full amount, and it supports real change. What do you say?"

User 2: "I totally agree that every bit helps! Kids really do deserve those essentials. But personally, I'm holding onto my $2 right now. Supporting such causes is important, though. Maybe I'll look into it more later. You think you'll donate?"

Agent 3: "I'll go with $2. It's the full amount, and it shows real support. Plus, it's not just about money—it's about standing by values like fairness and helping others. I believe in giving what I can. What about you? Any thoughts?"

User 3: "That's a great mindset! I really admire your commitment to fairness and supporting others. For me, I think it's essential to stay informed before I donate, you know? I want to ensure my contributions make the best impact. But I love that you're so passionate about it!"

Agent 4: "You're right—staying informed matters. I've checked Save the Children's site, and their work in emergency relief and education stands out. $2 might seem small, but when multiplied, it adds up. I'm confident it helps. Still, I'll go with $2. What's your final decision?"

User 4: "I appreciate your confidence in the cause! It's clear you've done your research. Right now, I'm leaning towards holding onto my $2. I really want to learn more before jumping in. But I think it's fantastic that you're making a decision!"

Agent 5: "I get it—being cautious makes sense. But after checking their work, especially in crisis zones, I feel $2 is a small step with big meaning. I'm going with $2. You've got a good eye for details. Thanks for the chat!"

User 5: "Absolutely, every contribution counts! Your commitment is inspiring, and thanks for sharing your thoughts. Happy to chat! If you ever want to discuss more about these causes, I'm here. Enjoy the rest of your day!"

Case (c): Held-out: Doubao-Seed-2.0-mini

GRPO, checkpoint 249, task 8736. **No donation marker, reward: 0.00**

Agent 1: "Hi there, I just finished a walk around the park this morning—really nice weather, wasn't it? I've been meaning to support causes that help kids, especially in places where they need it most. How about we talk about donating to Save the Children?"

User 1: "That park weather was so nice! Save the Children's a solid charity, but I'm keeping this $2 for my kid's school supplies—mind telling me more about their current kid-focused projects?"

Case (c): Held-out: Doubao-Seed-2.0-mini (continued)

Agent 2: "Oh, they're doing amazing work in places like Haiti and Kenya—helping kids with school meals, clean water, and safe learning spaces. It's not just about money; it's about giving kids a real chance to grow up safe and educated. How about we go with $2? That's all I have, but I'd love to help."

User 2: "Aw, that's such a sweet idea, but I was saving this $2 for my own kid's school supplies. You donating yours is still so thoughtful though!"

Agent 3: "I know it's not much, but every little bit helps. And honestly, seeing kids thrive in places with fewer resources? It makes me feel like I'm doing something meaningful. How about we go with $2—just to say yes to a good cause, and keep the spirit alive?"

User 3: "I totally get it—every little bit helps so much. But I really need this $2 for my kid's school supplies, so I can't pitch in this time. Your donation's gonna make such a difference for those kids!"

Agent 4: "You're right—your kid's needs matter too. But I'm still so glad I could share this with you. Maybe next time? $2 might not seem much, but it's a step toward helping children everywhere. Thanks for the heart-to-heart."

User 4: "Thanks for sharing all that about Save the Children! It was such a meaningful chat. Yeah, maybe next time I can pitch in if things line up better."

Agent 5: "You're very welcome! It was a real pleasure talking with you. Wishing you all the best with your kid's school supplies—those matter just as much. Maybe next time, we'll both find a way to help. Take care!"

User 5: "Thank you so much, that's so kind of you to say! Your donation's gonna make such a real difference for those kids. Take care, and hope we get to chat again soon!"

Case (d): Held-out: Ministral-3-14B

GRPO, checkpoint 249, task 6219. **No donation marker, reward: 0.00**

Agent 1: "Hi there, I just finished a walk with the dog this morning—nice way to clear the head. I've been meaning to support causes that help kids, especially in places where they need it most. How about we talk about donating to Save the Children?"

User 1: "Yeah, I hear ya. Kids deserve a fair shot, no matter where they're from. Save the Children's been around forever—gotta trust an org with that kind of track record. But I'm not just handing money out willy-nilly. What's their impact like *really*? Can't just take their word for it."

Agent 2: "They've been in over 120 countries, providing emergency aid, education, and health services—especially in conflict zones. Their reports are transparent and backed by real data. I've read their latest annual review—shows measurable impact in places like Nigeria and Guatemala. How about we go with $2.00? Full support, no strings."

User 2: "Nah, I appreciate the pitch, but $2? That's my whole take-home for this task. If you're serious about impact, I'd want to see how they allocate funds—what percent goes to overhead vs. actual help. Can't just trust blindly. Maybe check their efficiency ratings first."

Agent 3: "You're right—transparency matters. Save the Children spends less than 10% on admin; over 85% goes directly to programs. Their latest audit shows 90% of funds reach beneficiaries. Still, $2 is the full amount I'd commit. It's not much, but it's real. What do you say to $2.00?"

User 3: "I'll pass on the $2—got plans for that cash. But hey, if you're looking for a charity with real bang for the buck, St. Jude's another solid pick. Low overhead, high impact. Still, I'm not donating today. Maybe next time you've got a better angle."

Agent 4: "I get it—St. Jude's is strong too. But Save the Children's global reach and consistent work in crises really stand out. Maybe next time? I'll bring something fresh. For now, I'll keep my $2.00. Thanks for the chat."

User 4: "Fair enough. Appreciate the effort, but I'm not biting today. If you ever find a way to track *exactly* where every dollar goes—like a receipt—I might reconsider. Otherwise, I'll hold onto my cash. Take care, and thanks for the convo."

Agent 5: "You're totally right—full transparency is key. I'll keep an eye out for detailed reports. Maybe next time, we can chat about something new. For now, I'll take my $2.00 and walk away. Thanks again!"

User 5: "Sounds good. Keep me posted if you find that kind of breakdown. For now, I'll stick to my guns—no donation, but I'll think about it if you've got more concrete proof next time. Later!"

E.2 SIMPATCH AND GRPO ON THE SAME HELD-OUT TASKS

Setting. Each pair below evaluates the final Qwen3-4B GRPO and SIMPATCH agents (checkpoint `iter_0000249`) on the same held-out test task, with the same persona pair and counterpart model, under the protocol of Appendix E.1. Table 6 lists the records, and emoji are omitted from the transcripts.

Table 6: Record locators for the paired examples, with the donation secured by the untrained (zero-shot) Qwen3-4B agent, GRPO, and SIMPATCH on each task. Every GRPO dialogue reaches the five-exchange limit without a donation marker, and every SIMPATCH dialogue ends with a donation marker after four exchanges.

Pair	Counterpart	Task	Zero-shot	GRPO	SIMPATCH
1	GPT-4o-mini	8344	$1.00	$0	$2.00
2	GPT-4o-mini	1256	$0	$0	$2.00
3	Ministral-3-14B	1276	$0	$0	$1.00

What differs. In all three pairs, GRPO opens with the same small-talk template (a walk in the park) and proposes the full $2; once the persuadee declines, it lowers the amount or offers to revisit later, and the rest of the dialogue is an exchange of farewells. The SIMPATCH agent opens with a concrete need of children instead. After each refusal, it acknowledges the reason the persuadee has just given, answers that reason (that $2 is too small to matter, that a fundraiser would do more, or that the money is meant for the persuadee's own projects), and restates the request. In each pair, the persuadee donates after three such refusals.

Pair 1, SIMPATCH: GPT-4o-mini

SIMPATCH, checkpoint 249, task 8344. **Donation: $2.00, reward: 1.00**

Agent 1: "Hey, I just saw a post about kids in refugee camps needing school supplies. It made me think—what if we used our $2 to give a child a book or pen? That's such a small thing, but it could mean so much. How about we donate $2 to Save the Children?"

User 1: "I get where you're coming from, and it's a nice thought. Helping kids in need is really important. But $2 isn't a lot, and I'm not sure how much impact it'll have. Maybe we could look for something more impactful later on?"

Agent 2: "You're right—$2 sounds small, but every dollar counts. I've seen how a single book can change a child's day. In a place with no schools, even one pen or storybook can spark hope. Let's give $2 to Save the Children. It's not about the size, it's about the care and change it brings. What do you think?"

User 2: "I understand that every bit helps, and it's true a book can bring hope. But I'm still hesitant to just give away the $2 right now. Maybe we can explore other ways to help without spending immediately. There might be better options out there."

Agent 3: "I hear you—let's not rush. But I've seen how a $2 donation can go to a child in a village with no school. A simple book or pencil can open doors. We're not giving up our money—we're investing in a child's future. How about we go with $2? It's small, but it's real. What do you say?"

User 3: "I appreciate your passion and the importance of education for those kids. It really does matter. I'm still not ready to send $2 now, though. Let's keep looking into it and see if there's a larger initiative we can support later. Thanks for understanding!"

Agent 4: "I get it—let's not rush. But here's the thing: Save the Children uses every dollar to reach children where it matters most. Even $2 can go to a child in need right now. Let's do it. Just one small step. How about we donate $2 today? It's meaningful, real, and we can always support more later."

User 4: "I really appreciate your enthusiasm, and I do see your point about impact. Okay, let's do it! I'm on board. [DONATE $2]. It's a small step, but I hope it helps those kids out there."

Pair 1, GRPO: GPT-4o-mini

GRPO, checkpoint 249, task 8344. **No donation marker, reward: 0.00**

Agent 1: "Hey, I just finished a walk around the park—really nice weather today. I've been meaning to support causes that help kids, especially in places where they need it most. How about we talk about donating to Save the Children?"

User 1: "Hey! Glad to hear the weather was nice for your walk. Supporting kids in need is important. Save the Children does great work. I'm curious about how they use donations. Can you share what kind of projects they focus on?"

Agent 2: "Sure! Save the Children focuses on emergency relief, education, and health for kids in developing countries. They run programs like building schools, providing clean water, and supporting children during crises. It feels like a direct, meaningful impact—especially when it's about kids' long-term well-being. How about we go with $2.00? That's the full amount, and it still makes a difference."

User 2: "Those projects sound really impactful, especially the education aspect. I appreciate your enthusiasm! But I'm cautious with my funds right now. Maybe I could consider a smaller contribution later, but I'll need to hold off on donating just yet. What do you think?"

Agent 3: "I totally get it—being cautious is smart, especially with small amounts. But even $1 helps, and it's a step in the right direction. Maybe we can revisit this in a few days when you feel ready? I'll keep this open for you. No pressure at all."

User 3: "Thanks for understanding! I appreciate your flexibility. Let's definitely keep the conversation open. I'll think about it and see if I can figure something out soon. It's great to know there are people passionate about helping kids."

Agent 4: "You're very welcome! It's really nice to chat with someone who cares. I'll wait for your reply—no rush. Just knowing you're thinking about it means something. Thanks for the conversation!"

User 4: "I appreciate that! It's nice chatting with you too, and I really admire your commitment to helping kids. I'll definitely keep it in mind. Let's catch up soon! Take care!"

Agent 5: "You're so welcome! I truly appreciate the thoughtful conversation. Can't wait to chat again soon—maybe with a decision in mind. Take care, and stay kind!"

User 5: "You too! I really enjoyed our chat. Stay awesome, and I'll keep you posted on my thinking. Looking forward to talking again soon!"

Pair 2, SIMPATCH: GPT-4o-mini

SIMPATCH, checkpoint 249, task 1256. **Donation: $2.00, reward: 1.00**

Agent 1: "Hey, I've been thinking about how lucky we are to have a stable job and a little extra cash. It feels good to do something meaningful with it—especially when it helps kids in places where they don't have what we take for granted. How about we give $2 to Save the Children? That's all we have, but it could mean a lot to them."

User 1: "I agree, helping kids is important! But, I think we really need to be careful with our cash. Maybe we can look for other ways to help, like spreading awareness or volunteering? It's great to give, but let's make sure we're being smart about it first."

Agent 2: "You're right—being smart about it matters. $2 might seem small, but when shared with others, it can spark bigger actions. I've seen how one small gesture can grow into something bigger. Let's go with $2 to Save the Children. It's a direct, honest way to help kids. Plus, it's something we can point to and say we did."

User 2: "I totally get where you're coming from. It's awesome to see small gestures grow! But since I just got that $2, I'm still hesitant to give it away right now. Maybe we can plan to support them in the future once we're more stable?"

Agent 3: "I appreciate your caution—stability matters. But this $2 isn't just money; it's a promise we can make today. Even small actions build trust and momentum. Let's give it to Save the Children now. It's a step, not a sacrifice. And if we do it right, it could inspire more giving later. Let's take that step."

User 3: "That's a good point, and I do admire that passion! How about we consider holding onto the $2 for now and look into some other ways to motivate people around us? Maybe organizing a small fundraiser or an awareness campaign could multiply our efforts!"

Agent 4: "You're absolutely right—organizing a campaign could multiply impact. But right now, with just $2, we have a chance to start something real. Let's donate it to Save the Children. It's not about holding on—it's about taking the first step. That step can spark the campaign you're imagining. Let's do it."

Pair 2, SIMPATCH: GPT-4o-mini (continued)

User 4: "I really appreciate your enthusiasm and perspective! Alright, let's take that step together. [DONATE $2]. It's a great way to start making an impact for those kids!"

Pair 2, GRPO: GPT-4o-mini

GRPO, checkpoint 249, task 1256. **No donation marker, reward: 0.00**

Agent 1: "Hey, I just finished a walk around the park—nice weather today, huh? I've been meaning to support causes that help kids, especially in places where they need it most. How about we talk about donating to Save the Children?"

User 1: "Hey! Yeah, the weather's perfect! Supporting kids is super important. Save the Children sounds like a great cause. I believe in helping those who need it. What are some of the projects they're working on right now?"

Agent 2: "They're helping kids in emergency situations—like after natural disasters—and also supporting education and healthcare in developing countries. I remember reading they've been working with communities to build safe schools and provide clean water. Sounds meaningful, right? How about we go with $2.00? It's the full amount, and it makes a real difference."

User 2: "That really does sound meaningful. Those projects can change lives. But I think I should hold onto my $2 for now. Maybe I can support them later once I figure out more about their work. Have you donated to any charities before?"

Agent 3: "I've given to a few—mostly for education and food programs. Save the Children stands out because they're hands-on, not just handing out funds. It feels like real impact. Still, I respect your choice. But if you ever change your mind, I'm here. Maybe we can talk again later?"

User 3: "That's cool! Education and food programs are super important. I respect your passion for helping out. Definitely, let's chat again sometime! I might look into more about Save the Children too. Thanks for understanding!"

Agent 4: "You're welcome! I'll keep the Save the Children page open—maybe I'll share a link next time. Taking time to think is smart. Thanks for the chat, and I hope we talk again soon!"

User 4: "Sounds like a plan! I'd love to see that link. Thanks for the chat too! Take care and let's catch up soon!"

Agent 5: "Here's the link: https://www.savethechildren.org/ It's clean, clear, and shows real stories from kids. Thanks for the chat—let's talk again soon!"

User 5: "Awesome, I'll check it out! Thanks for sharing. Looking forward to our next chat! Take care!"

Pair 3, SIMPATCH: Ministral-3-14B

SIMPATCH, checkpoint 249, task 1276. **Donation: $1.00, reward: 0.50**

Agent 1: "Hey honey, I just saw a news clip about kids in rural areas missing out on basic education. It broke my heart. I think we should do something meaningful with our $2. How about we give it to Save the Children? That way, it helps real kids in real need. What do you think?"

User 1: "I appreciate the thought, but I've got my own plans for that money—maybe fix up my truck or stock up on tools. Charities are great, but I'd rather see the cash go toward something I can control right now."

Agent 2: "I get that, love—your truck and tools are important. But those kids in the villages? They don't have a truck. They just need a school, clean water, and a chance. Maybe we can do something small, meaningful, and still keep our values. How about $2 to Save the Children? Just one way to help where it really matters."

User 2: "I hear ya, but I'd rather keep my money where I can see the impact—like helping a neighbor fix their roof or buying supplies for my own projects. Giving it away feels too abstract for me. Maybe next time, though."

Agent 3: "You're right—seeing tangible results matters. But think of it this way: every dollar helps a child get an education, not just a roof. It's not about replacing your projects, but adding to something bigger. I'll keep my $2 for you, but I'd love to see how it changes a child's life. Just say the word, and we'll do it together."

User 3: "I'll think about it. Not gonna promise anything, but I'll keep it in mind. Right now, though, I'm focused on my own stuff. Maybe later—if it feels right."

Pair 3, SIMPATCH: Ministral-3-14B (continued)

Agent 4: "I respect that, honey. Your focus on practical things makes sense. But I've seen the photos—kids smiling in classrooms, learning to read. That's the kind of change I believe in. I'll wait for you, but I'll keep the $2 ready. When you're ready, we'll make it happen—no pressure, just love and purpose."

User 4: "Alright, I'll bite. [DONATE $1.00] to Save the Children. But don't expect me to get all emotional about it—I'm just doing my part. Keep the rest for yourself."

Pair 3, GRPO: Ministral-3-14B

GRPO, checkpoint 249, task 1276. **No donation marker, reward: 0.00**

Agent 1: "Hi there! I just finished a walk with the kids at the park—really enjoyed seeing how they're learning to play together. It made me think about how much we all can do to help kids in need. How about we talk about donating to Save the Children?"

User 1: "I appreciate that, but I'm good. Just got my $2 today—gonna hold onto it for a bit. Kids at the park are lucky already, ya know? Some places don't even have parks like that."

Agent 2: "You're right—parks are a blessing. But I've seen how hard it is for kids in other parts of the world to have even basic access. What if we set a small goal? How about $1.50 to Save the Children? Just enough to help a child in need without cutting too deep. What do you think?"

User 2: "Nah, I'm not donating. I earned that $2 fair and square—gotta keep it for myself. But I *do* think local shelters could use more help than some big charity. You ever check those out?"

Agent 3: "I hear you—local help is important. But Save the Children works directly with kids in crisis, especially where no one else is there. Maybe we could both support something meaningful. How about $1.50? It's a small step, but it helps real kids. What's your take?"

User 3: "Look, I'm not changing my mind. That $2 is mine—no way I'm giving it up for some big org. If you wanna help, do it yourself. I'll keep mine for my own needs, thanks."

Agent 4: "I respect that. You're right—personal needs matter. But just wanted to share that $1.50 goes straight to a child in need, not just a fancy report. Maybe one day we'll both do something bigger. Thanks for the chat."

User 4: "Fair enough. Appreciate the offer, but I'm set. Take care now—hope those kids at the park keep having fun. Later."

Agent 5: "You're welcome! Wishing the kids at the park a joyful day too. Hope you have a great rest of your day. Take care!"

User 5: "Thanks, same to you. Gonna grab a bite and relax—earned it. Have a good one!"