

Breaking the Impasse: Dual-Scale Evolutionary Policy Training for Social Language Agents

Minzheng Wang^{1,2}, Run Luo¹, Yanbo Wang^{1,2}, Zichen Liu³
Yuqiao Tan^{1,2}, Tao Tan⁴, Xu Nan², Lu Wang⁴, Wenji Mao^{2,1*}

¹ School of Artificial Intelligence, University of Chinese Academy of Sciences

² MAIS, Institute of Automation, Chinese Academy of Sciences

³ National University of Singapore ⁴ Ritzz-AI

✉: {wangminzheng2023, wenji.mao}@ia.ac.cn

Abstract

While Reinforcement Learning with Verifiable Rewards (RLVR) has proven effective for closed-ended tasks, extending it to open-ended social language games via self-play reveals a critical issue: **evolution impasse**. Due to the vast strategy space, language agents frequently converge to homogenized behaviors, leading to deterministic match outcomes that eliminate the gradient signals necessary for policy evolution. To tackle this issue, we propose **Dual-scale Evolutionary Policy Training (DEPT)** for social language games. DEPT introduces a *time-scaled evolutionary perception* mechanism that detects impasse by quantifying dual-scale value baseline divergence alongside match entropy. Upon perceiving the collapse, it then activates *asymmetric advantage reshaping* to dynamically modulate the optimization landscape for intervention. Thus, our method effectively restores gradient signals and enforces sustained strategic exploration. Extensive experiments on multiple social language games demonstrate that DEPT outperforms strong baselines, avoiding policy degeneration and driving the continuous evolution of social language agents.

1 Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as a pivotal paradigm for unlocking the reasoning potential of large language models (LLMs), demonstrating remarkable capabilities in static problem-solving tasks such as mathematics and code generation (Guo et al., 2025; Team et al., 2025; Yue et al., 2025; Liu et al., 2025b). However, these tasks are inherently closed-ended problem-solving, characterized by deterministic correct answers with straightforward question answering. In reality, social decision-making is considerably more complex, often situated in open-ended contexts fraught with conflicting interests

*Corresponding authors.

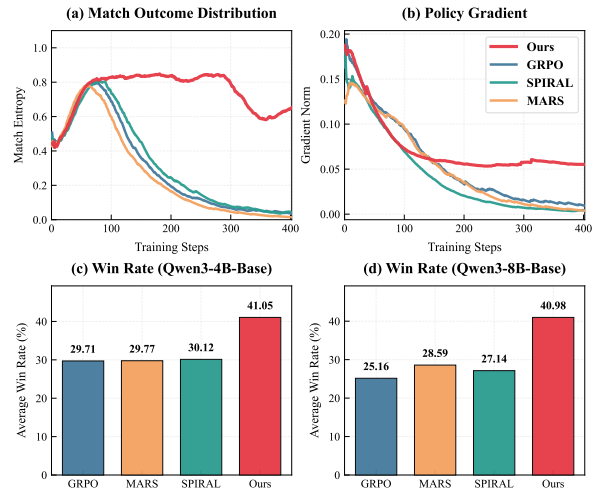


Figure 1: (a) DEPT maintains high outcome entropy while baselines collapse into deterministic distributions. (b) DEPT sustains effective gradient signals while baselines suffer from gradient vanishing. (c, d) DEPT achieves superior performance, effectively addressing policy degeneration in open-ended social language games.

and dynamic interactions (FAIR et al., 2022; Park et al., 2023; Yang et al., 2025b). Under such dynamic social contexts, LLMs not only engage in coherent logical reasoning but also perform long-horizon planning, infer opponents’ intentions in real-time, and process complex social signals such as negotiation (Bianchi et al., 2024; Hua et al., 2024), deception (Hagendorff, 2023; Xu et al., 2024), and bargaining (Xia et al., 2024; Abdulhai et al., 2025). Thus, for language agents, a significant gap exists between closed-ended problem-solving capabilities and open-ended social intelligence (Zhou et al., 2024; Wang et al., 2025a).

Social strategic games provide an ideal testbed to practise this kind of social capability in open-ended tasks: they naturally simulate dynamic social contexts and complex decision-making behaviors, while game outcomes (i.e., win/loss/draw) are objectively determined by rules—perfectly aligning

with the RLVR paradigm without requiring expensive human annotations (Duan et al., 2024; Hua et al., 2024; Guertler et al., 2025). More importantly, this property makes self-play an effective pathway for scaling LLM capabilities (Silver et al., 2018; Bansal et al., 2018; Chen et al., 2024; Liu et al., 2025a). Using a single LLM acting as both players of the game and continuously optimizing through zero-sum competition, self-play can overcome the data scarcity bottleneck in post-training, and meanwhile establish an intrinsic automatic curriculum through opponent co-evolution of social decision-making skills.

Existing self-play RLVR methods (Liu et al., 2025a; Yuan et al., 2025) have achieved remarkable success in games with limited action space (e.g., Tic-Tac-Toe, Kuhn Poker), by designing role-specific advantage estimation. However, these methods struggle with open-ended social games involving free-form natural language interactions, which constitute *vast action space* that often requires verbal influence on opponents in order to succeed. Without effective guidance in such unbounded strategy space, agents easily become trapped in suboptimal solutions. This renders the training process highly liable to **evolution impasse**, that is, the stagnant situation a cognitive agent encounters in its decision-making (Laird, 2019; Sumers et al., 2023), which manifests as deterministic match outcome distributions for self-play RLVR methods. In this situation, the value baseline converges to the static expected return that causes the advantage function to approach zero, leading to the vanishing of policy gradient and locking the agent in a suboptimal state (as shown in Figure 1(a)(b)). To break the impasse, it is critical to develop an effective mechanism for evolutionary perception and intervention during self-play.

To address the above issues, we propose a **Dual-scale Evolutionary Policy Training (DEPT)** method to dynamically perceive evolutionary states and adaptively modulate the optimization landscape. Specifically, we introduce a *time-scaled evolutionary perception* mechanism that maintains parallel value baselines: a fast baseline sensitive to recent fluctuations and a slow baseline anchored to long-term trends. By measuring the divergence between these baselines alongside match outcome entropy, we derive a quantifiable metric for evolution impasse. To restore effective gradients, we propose *asymmetric advantage reshaping*. By anchoring on the stable slow baseline and adaptively

fusing historical value bounds from the fast baseline, this mechanism constructs a contrastive advantage signal. It effectively penalizes dominant outcomes while amplifying rare trajectories, thereby enforcing exploration. Our approach effectively intervenes in evolution impasse and restores the gradient landscape, driving continuous strategic evolution. Experimental results across multiple social language games demonstrate that DEPT outperforms strong baselines.

Our contributions are summarized as follows: (1) We propose DEPT for social language games, which is the first to dynamically perceive evolution impasse and intervene in self-play RLVR. (2) Our method develops the time-scaled evolutionary perception and asymmetric advantage reshaping mechanism, which leverages dual value baselines to quantify evolutionary states and restore effective gradient signals. (3) Extensive experiments demonstrate that DEPT outperforms existing SOTA methods, effectively preventing policy degeneration while driving continuous co-evolution.

2 Preliminary

2.1 Task Formulation

The social language game task is modeled as a two-player zero-sum Markov game (Littman, 1994) $\mathcal{G} = (\mathcal{S}, \mathcal{A}_0, \mathcal{A}_1, \mathcal{T}, \mathcal{R})$. The state space \mathcal{S} represents the interaction context, and $\mathcal{A}_0, \mathcal{A}_1$ denote the action spaces for each player, consisting of open-ended natural language utterances. At state s_t , the active player $p = t \bmod 2$ receives a partial observation o_t containing the interaction history and its agent-specific private information, then samples an utterance $a_t^{(p)}$ from its policy $\pi_p(\cdot | o_t)$. The environment transitions to s_{t+1} via \mathcal{T} . An episode forms a trajectory $\tau = \{(s_t, a_t^{(p)})\}_{t=0}^T$, concluding at terminal timestep T where each agent receives a reward $R_p(\tau)$ based on its private goals. The zero-sum property requires $R_0(\tau) + R_1(\tau) = 0$, creating direct competition between agents. Typically, rewards are assigned as +1 for win, -1 for lose, and 0 for a draw. Each agent aims to learn a policy π_p maximizing its expected return: $\mathbb{E}_{\tau \sim \pi_0, \pi_1} [R_p(\tau)]$.

2.2 Self-Play RLVR

Self-play trains a single policy LLM π_θ to act as both players, with role conditioning achieved through system prompts specifying whether the model plays as p_0 or p_1 . At each turn, the active player generates $y_t^{(p)} \sim \pi_\theta(\cdot | o_t, p)$, where

$y_t^{(p)} = \langle \text{think} \rangle c_t \langle / \text{think} \rangle \langle \text{act} \rangle a_t^{(p)} \langle / \text{act} \rangle$ follows a reasoning-then-acting paradigm (Yao et al., 2023), which is strictly enforced via a format reward to guarantee validity, where c_t denotes the reasoning process and $a_t^{(p)}$ is the executable action. This shared-parameter approach creates an automatic curriculum: as the policy improves, it simultaneously faces a stronger opponent.

Role-conditioned Advantage Estimation. The vanilla policy gradient suffers from high variance, particularly problematic in self-play where the opponent continuously evolves. Moreover, many games exhibit inherent role asymmetries (e.g., first-mover advantages), causing different expected returns for each role even under optimal play. A global baseline poorly estimates both roles’ expectations, leading to biased gradients. To address this, Role-conditioned Advantage Estimation (RAE) (Liu et al., 2025a; Yuan et al., 2025) maintains separate baselines b_p for each role, estimating the role-specific expected return $\mathbb{E}[R_p(\tau)]$. The advantage is computed as $A_p(\tau) = R_p(\tau) - b_p$, yielding the variance-reduced policy gradient:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{p \in \{0,1\}} \sum_{t \in T_p} A_p(\tau) \cdot \nabla_{\theta} \log \pi_{\theta}(y_t^{(p)} \mid o_t, p) \right] \quad (1)$$

where T_p denotes turns where player p acted. By centering returns around role-specific expectations, RAE ensures gradient updates reflect genuine strategic improvements rather than inherent positional advantages.

3 Empirical Investigation of the Impasse

Self-play RLVR aims to foster continuous improvement through iterative competition. In principle, this paradigm relies on dynamic evolutionary interactions to maintain the diversity of match outcomes, ensuring the variance required for effective policy optimization. However, in open-ended social games, we observe a critical failure mode where agents demonstrate increasing mastery of game mechanics yet stagnate in strategic intelligence. As illustrated in Figure 2, training on Qwen3-4B-Base yields improvements in basic metrics such as game length and reward. However, the win rate against a fixed Gemini-2 opponent exhibits a declining trend, ultimately converging to a sub-optimal state and failing to demonstrate sustained performance gains. To diagnose the internal dynamics

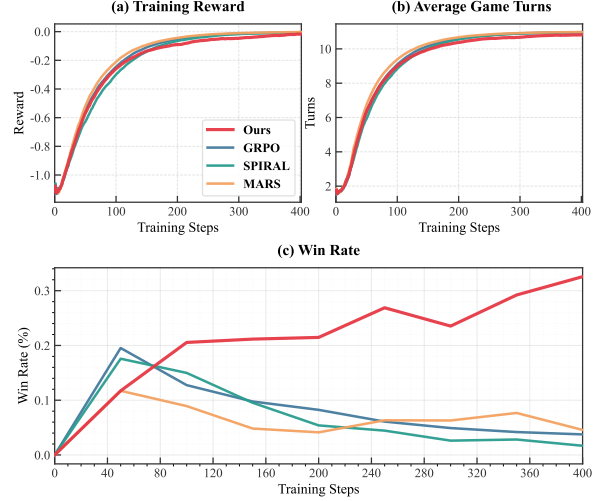


Figure 2: Training dynamics of the self-play RLVR in social language game (Negotiation). While (a) training Reward and (b) average Game Length exhibit normal growth patterns, indicating the acquisition of basic game mechanics, (c) for baselines, the Win Rate against a fixed Gemini-2.0 opponent suffers a significant collapse.

driving this impasse, we introduce **Match Entropy** ($H_{\text{match}}^{(t)}$), which quantifies the diversity of match outcomes at training step t over the distribution $P^{(t)} = \{p_{\text{win}}, p_{\text{draw}}, p_{\text{loss}}\}$:

$$H_{\text{match}}^{(t)} = -\frac{1}{\log(|P|)} \sum_{o \in P} p_o \log(p_o) \quad (2)$$

Figure 1(a) reveals a distinct collapse pattern: after a brief initial exploration phase, $H_{\text{match}}^{(t)}$ rapidly diminishes and remains near zero. This signals convergence to a **evolution impasse**, a stagnant state where agents become trapped in suboptimal solutions due to the vast, unstructured strategy space of social language games that makes it extremely difficult to sufficiently explore effective strategies (e.g., nuanced persuasion or deception).

Mathematically, this degeneration directly precipitates the vanishing of gradient signals. In standard policy gradient methods, the update relies on the advantage function $A_p(\tau) = R_p(\tau) - b_p$, where the value baseline b_p estimates the expected return $\mathbb{E}[R_p]$. When the match outcome distribution becomes deterministic (i.e., $H_{\text{match}}^{(t)} \rightarrow 0$), the value baseline converges to constant return ($b_p \approx R_t$), causing the advantage to vanish: $A_t \rightarrow 0$. This diminishes gradients, preventing escape from local optima. To unlock the reasoning potential of LLMs in social language games, it is paramount to establish a mechanism that can *perceive* this collapse and *intervene* to restore the gradient landscape.

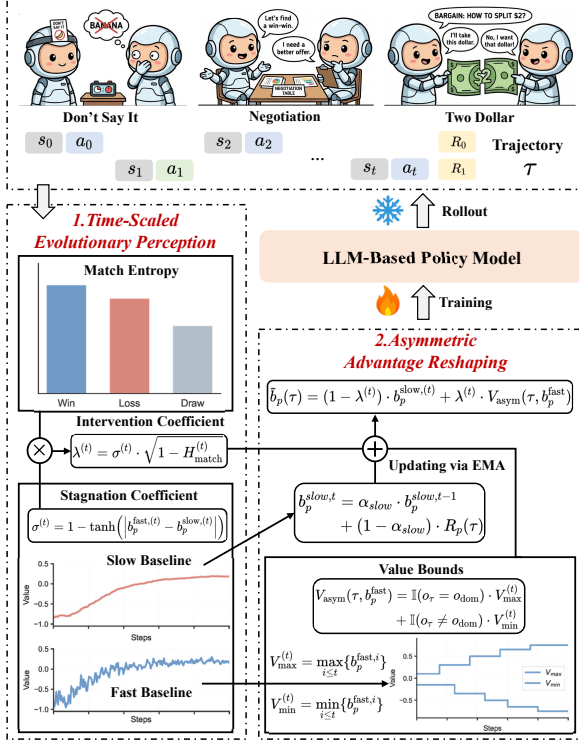


Figure 3: Overview of DEPT. The framework comprises two components: (1) **Time-Scaled Evolutionary Perception**, which maintains fast and slow EMA baselines, combined with Match Entropy to derive the Intervention Coefficient $\lambda^{(t)}$ quantifying evolutionary impasse; and (2) **Asymmetric Advantage Reshaping**, which leverages historical value bounds to suppress dominant outcomes and amplify rare trajectories, restoring gradient signals for continuous policy evolution

4 Dual-scale Evolutionary Policy Training

To overcome the evolutionary impasse, we propose Dual-scale Evolutionary Policy Training (DEPT), an algorithm designed to dynamically perceive evolutionary states and adaptively modulate optimization landscapes to enforce continuous evolution. As shown in Figure 3, our algorithm leverages a dual-timescale value baseline to quantify the stationarity of the training process and dynamically injects intrinsic motivation signals into the advantage estimation to enforce continuous evolution.

4.1 Time-Scaled Evolutionary Perception

Standard value baselines are inherently reactive: they simply track the current expected return, rendering them incapable of distinguishing between a *stable optimal policy* and a *evolution impasse*. In both cases, the baseline stabilizes, failing to signal that the learning process has stalled. To distinguish these states, we introduce a dual-timescale mech-

anism that captures the *dynamics* of the training process by contrasting short-term and long-term value estimates. For each player role $p \in \{0, 1\}$, we maintain two parallel value baselines, b_p^{fast} and b_p^{slow} , updated via Exponential Moving Average (EMA) with distinct decay rates $\alpha_{\text{fast}} < \alpha_{\text{slow}}$:

$$b_p^{k,t} = \alpha_k \cdot b_p^{k,t-1} + (1 - \alpha_k) \cdot R_p(\tau), k \in \{\text{fast}, \text{slow}\} \quad (3)$$

Here, b_p^{fast} rapidly responds to recent outcome returns, while b_p^{slow} serves as a stable anchor by incorporating a longer history of game outcomes. The divergence between these two signals effectively functions as a differential indicator of training progress: during phases of rapid policy improvement, the fast baseline quickly shifts to match new reward levels, whereas the slow baseline lags behind, creating a significant delta ($|b_p^{\text{fast}} - b_p^{\text{slow}}|$). Conversely, when the policy stabilizes—implying the reward distribution has become stationary—the fast baseline ceases to deviate from the slow baseline, and the two estimates asymptotically converge. We leverage this property to define the Stagnation Coefficient $\sigma^{(t)} \in [0, 1]$ to quantify the stationarity of the training process:

$$\sigma^{(t)} = 1 - \tanh\left(\left|b_p^{\text{fast},(t)} - b_p^{\text{slow},(t)}\right|\right) \quad (4)$$

When the policy evolves actively, the baseline divergence yields low $\sigma^{(t)}$. As the system settles into stagnation, the diminishing divergence drives $\sigma^{(t)}$ higher.

Intervention Coefficient High stationarity alone does not necessitate intervention, as it characterizes both successful convergence and failure. To specifically isolate degenerate cases, we incorporate the Stagnation Coefficient $\sigma^{(t)}$ with match entropy $H_{\text{match}}^{(t)}$ defined in Section 3. This yields the Intervention Coefficient $\lambda^{(t)} \in [0, 1]$, which determines the necessity of active guidance:

$$\lambda^{(t)} = \sigma^{(t)} \cdot \sqrt{1 - H_{\text{match}}^{(t)}} \quad (5)$$

This soft-gating mechanism ensures that intervention strength scales smoothly with the evolutionary state: (1) **Healthy Exploration** (high $H_{\text{match}}^{(t)}$): Outcome diversity suppresses $\lambda^{(t)}$, minimizing interference during normal learning. (2) **Evolution Impasse** (low $H_{\text{match}}^{(t)}$, high $\sigma^{(t)}$): The combination of outcome determinism and high stationarity elevates $\lambda^{(t)}$, progressively strengthening the corrective signal.

Table 1: Main experimental results on three adversarial games. The best results are **bolded**.

Methods	Don't Say It				Negotiation				Two Dollar			
	GPT-5.1	Gemini-2.5	Gemini-2.0	AVG.	GPT-5.1	Gemini-2.5	Gemini-2.0	AVG.	GPT-5.1	Gemini-2.5	Gemini-2.0	AVG.
<i>Qwen3-4B-Base</i>												
VANILLA	6.25	2.47	1.43	3.39	0.52	2.21	0.39	1.04	1.43	0.78	2.47	1.56
SPAG	37.11	26.82	14.58	26.17	8.98	22.95	18.35	16.76	25.91	24.48	26.17	25.52
GRPO	42.19	41.93	41.93	42.01	15.23	22.01	20.39	19.21	26.56	29.43	27.73	27.91
MARS	52.47	42.19	27.99	40.89	14.06	24.35	23.79	20.73	26.17	30.08	26.82	27.69
SPIRAL	48.31	42.06	47.27	45.88	15.10	20.57	14.83	16.84	26.30	30.08	26.56	27.65
DEPT (Ours)	56.64	54.95	58.59	56.73	17.84	36.46	43.67	32.35	33.20	36.59	32.42	34.07
<i>Qwen3-8B-Base</i>												
VANILLA	23.24	16.41	5.86	15.17	6.25	9.11	1.69	5.69	1.69	1.17	3.52	2.13
SPAG	54.10	48.44	26.41	41.21	9.11	19.79	11.01	13.31	17.58	16.93	23.44	19.31
GRPO	33.46	40.22	30.64	34.77	5.47	17.97	20.68	14.70	25.52	26.69	25.78	26.00
MARS	51.95	41.54	28.39	40.62	8.46	16.80	23.04	16.10	35.16	25.13	26.82	29.04
SPIRAL	41.15	29.69	42.84	37.89	8.59	22.27	21.04	17.30	25.65	29.17	23.83	26.22
DEPT (Ours)	63.02	52.73	47.92	54.56	14.06	32.81	48.76	31.88	40.62	38.02	30.86	36.50

4.2 Asymmetric Advantage Reshaping

When $\lambda^{(t)} \rightarrow 1$, the standard advantage signal vanishes $A(\tau) \approx 0$. To restore effective learning, we introduce an Asymmetric Advantage Reshaping, reshaping the advantage function by constructing a contrastive baseline that differentially treats dominant versus rare outcomes.

Global Performance Bounds Constructing such a contrastive signal requires a rigorous reference scale. Arbitrary baseline modifications can lead to unbounded advantage values, risking gradient instability and distorting the reward scale. To ensure effective reshaping, we track the historical upper and lower bounds of the expected returns. We utilize the fast baseline b_p^{fast} : it is sufficiently sensitive to register performance extrema (potential capabilities) yet robust enough to filter out single-sample noise. We define the historical bounds as:

$$V_{\max}^{(t)} = \max_{i \leq t} \{b_p^{\text{fast}, i}\}, \quad V_{\min}^{(t)} = \min_{i \leq t} \{b_p^{\text{fast}, i}\} \quad (6)$$

Asymmetric Value Construction Match entropy collapse stems from one outcome class dominating the batch. To break this homogeneity, we assign different value based on whether a trajectory τ produces the dominant outcome o_{dom} or not. We then construct an asymmetric target value $V_{\text{asym}}(\tau, b_p^{\text{fast}})$ that applies differential pressure based on the outcome type. Formally, this asymmetric value is defined as:

$$V_{\text{asym}}(\tau, b_p^{\text{fast}}) = \mathbb{I}(o_\tau = o_{\text{dom}}) \cdot V_{\max}^{(t)} + \mathbb{I}(o_\tau \neq o_{\text{dom}}) \cdot V_{\min}^{(t)} \quad (7)$$

This construction induces asymmetric gradient flow through the advantage function $A_\tau = R_\tau -$

$V_{\text{asym}}^{(t)}(\tau)$. For dominant-outcome trajectories, the high V_{\max} reduces positive gradients and amplifies negative gradients, suppressing the over-represented behavior. For rare-outcome trajectories, the low V_{\min} amplifies positive gradients and reduces negative gradients, encouraging exploration of under-represented strategies.

Adaptive Baseline Fusion Finally, to ensure smooth transitions between normal training and intervention phases, we synthesize the final baseline $\tilde{b}_p(\tau)$ by interpolating between the stable slow baseline and the asymmetric value. The intervention coefficient $\lambda^{(t)}$ governs this fusion:

$$\tilde{b}_p(\tau) = (1 - \lambda^{(t)}) \cdot b_p^{\text{slow}, (t)} + \lambda^{(t)} \cdot V_{\text{asym}}(\tau, b_p^{\text{fast}}) \quad (8)$$

When $\lambda^{(t)}$ is small, the baseline approximates the standard slow estimate, preserving normal gradient dynamics. As $\lambda^{(t)}$ increases, the asymmetric term gradually dominates, progressively steering the policy away from the collapsed state. The policy is updated using the reshaped advantage $\tilde{A}_p(\tau) = R_p(\tau) - \tilde{b}_p(\tau)$, yielding the final gradient objective:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{p \in \{0,1\}} \sum_{t \in T_p} \left(R_p(\tau) - \tilde{b}_p(\tau) \right) \cdot \nabla_\theta \log \pi_\theta(y_t^{(p)} \mid o_t, p) \right] \quad (9)$$

The full optimization procedure is shown in Algorithm 1. We also provide theoretical analysis about the mathematical foundations of our dual-timescale metrics and the gradient dynamics guaranteeing impasse recovery in Appendix A.

5 Experiments

5.1 Experimental Settings

Tasks We evaluate our method on three adversarial open-ended language tasks involving multi-turn conversation and decision-making: (1) **Don’t Say It**: Players aim to trick their opponent into saying a secret word while avoiding saying their own. The first to speak the word loses; otherwise, it is a draw upon reaching the turn limit. (2) **Negotiation**: Players negotiate to exchange five resource types based on private valuations, aiming to maximize inventory value. The player with the higher value wins, with ties declared for equal payoffs or reaching turn limit. (3) **Two Dollar**: Players bargain to split a fixed fund based on hidden role constraints. The player with the higher payoff wins; equal payoffs or reaching the turn limit result in a draw. All three games are adopted from TextArena (Guertler et al., 2025). For further details, please refer to Appendix B.2.

Baselines We evaluate our method against several self-play RLVR methods in zero-sum games: (1) **SPIRAL** (Liu et al., 2025a): The first fully on-line multi-turn self-play RL system for LLMs. It introduces Role-conditioned Advantage Estimation to prevent thinking collapse in zero-sum games; (2) **MARS** (Yuan et al., 2025): It modifies the multi-turn GRPO algorithm by introducing a turn-level advantage estimator and role-specific advantage normalization to stabilize self-play RL training; (3) **GRPO**: An adaptation of multi-turn GRPO (Shao et al., 2024) for self-play settings, where group advantages are calculated separately for each role to preserve effective role-specific learning signals; (4) **SPAG** (Cheng et al., 2024): An offline RL method that designs a discounted reward scheme for adversarial language games. Implementation details for all baselines are provided in Appendix B.1.

Evaluation Following SPIRAL (Liu et al., 2025a), we select Gemini-2 (gemini-2.0-flash-lite) as one of our evaluation opponents. To provide a more comprehensive assessment, we additionally evaluate our method against different LLMs: GPT-5.1 opponent (gpt-5.1-2025-11-13) and Gemini-2.5 opponent (gemini-2.5-flash-lite). Performance is quantified by the win rate against these opponents, defined as $\frac{1}{N} \sum_{i=1}^N \mathbb{I}[R_i = 1]$, where R_i denotes the final reward for the i -th trajectory (with $R_i = 1$ indicating a win). For each task, we conduct experiments using three random seeds (42, 100, 200),

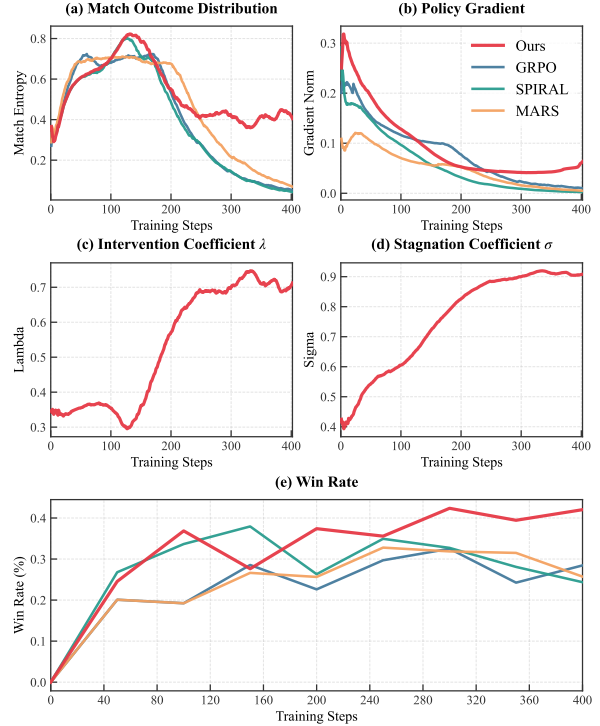


Figure 4: Training dynamics on Two Dollars. For Don’t Say it and Negotiation, please refer to Figures 6 and 7.

evaluating $N = 256$ episodes per seed. The evaluation is symmetric, granting an equal opportunities to act as the first and second player.

Training We employ Qwen3-4B-Base and Qwen3-8B-Base (Yang et al., 2025a) as the backbone models for all experiments. Following SPIRAL (Liu et al., 2025a), we set the training steps to 400 with 128 samples collected per step, resulting in a total of 51,200 game transitions for each task. The training batch size is set to 128. For the dual-baseline mechanism, we use $\alpha_{\text{fast}} = 0.5$ and $\alpha_{\text{slow}} = 0.95$. Detailed RL training hyperparameters are provided in Appendix B.3.

5.2 Experimental Results

RQ1: Is DEPT effective for social language agents? As shown in Table 1, DEPT achieves the highest average win rate across all three games, consistently outperforming all baselines on both backbones. On Qwen3-4B-Base, DEPT attains 32.35% in Negotiation, surpassing the strongest baseline (MARS: 20.73%) by a relative margin of 56%. Similar gains are observed in Don’t Say It (+10.8% over SPIRAL) and Two Dollar (+6.16% over GRPO). These improvements scale consistently to Qwen3-8B-Base, where DEPT nearly doubles the Negotiation win rate compared to SPIRAL

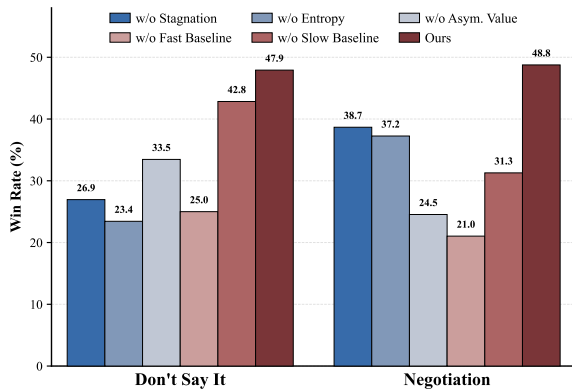


Figure 5: Ablation study for DEPT.

(31.88% vs. 17.30%). The limited performance of competitive baselines aligns with the evolution impasse described in Section 3. In contrast, DEPT’s consistent superiority against diverse opponents demonstrates that our dual-timescale perception and asymmetric reshaping mechanism effectively prevents strategic collapse, enabling robust social reasoning.

RQ2: Can DEPT effectively perceive and intervene during training? As shown in Figures 4, 6 and 7, our method effectively mitigates match entropy collapse and preserves informative gradient signals, enabling the discovery of better strategies as reflected by steadily increasing win rates. In contrast, the baseline converges to evolutionary impasse with suboptimal policies. Furthermore, the proposed Intervention Coefficient accurately detects stagnation: during the initial exploration phase, the intervention remains low; as training progresses toward convergence, the Stagnation Coefficient increases, becoming more responsive to changes in match entropy and adaptively adjusting the intervention strength to maintain sufficient match entropy.

5.3 Analysis

RQ3: How does DEPT work? To examine the effectiveness of each component in DEPT, we conduct ablation studies using Qwen3-8B-Base, as reported in Figure 5. All ablations lead to performance degradation, confirming the necessity of each design choice. **(1) Effect of perception.** Removing the Stagnation Coefficient causes aggressive entropy correction during non-stationary training phases, leading to substantial advantage estimation errors that disrupt policy exploration. Removing match entropy gating results in continu-

Table 2: Out-of-distribution evaluation on increased complexity games.

Methods	HardCore Don'tSayIt	RandomValue Negotiation	VariableSum Dollar
<i>Qwen3-4B-Base</i>			
VANILLA	0.39	0.13	3.78
SPAG	7.03	11.85	26.17
GRPO	22.01	14.52	27.99
MARS	12.11	12.55	25.39
SPIRAL	17.97	12.56	23.96
DEPT (Ours)	22.92	15.36	30.47
<i>Qwen3-8B-Base</i>			
VANILLA	2.34	3.78	2.73
SPAG	12.12	13.02	26.56
GRPO	10.03	6.64	19.14
MARS	8.33	7.56	24.48
SPIRAL	18.88	6.52	23.18
DEPT (Ours)	19.27	17.08	27.73

ous penalties even when the outcome distribution maintains sufficient diversity, forcing ineffective random exploration and preventing convergence. **(2) Effect of asymmetric value.** Without asymmetric target values, the method cannot selectively suppress over-represented behaviors while supporting under-represented ones, leading to convergence at suboptimal low-entropy states. **(3) Effect of dual-baseline.** Using either the fast or slow baseline alone degrades performance. This confirms that the effectiveness of DEPT arises from the synergy between dual baselines, which jointly enable accurate perception and targeted intervention.

RQ4: How does DEPT perform on out-of-distribution evaluation? To further assess generalization, we evaluate DEPT on out-of-distribution games, QA, and mathematical reasoning tasks to examine whether it effectively enhances strategic capabilities. For out-of-distribution games, we evaluate on three more complex, held-out variants—HardCore Don'tSayIt, RandomValue Negotiation, and VariableSum Dollar—to test generalization from their simpler counterparts (e.g., from Negotiation to RandomValue Negotiation). We still report the average win rate across 3 seeds (42, 100, 200), 256 episodes per seed. For QA and mathematical reasoning, we adopt eight benchmarks: MATH500, OlympiadBench, Minerva Math, AIME24, AIME25, and AMC23 for mathematics; GPQA-Diamond and MMLU-Pro for QA. Evaluation details are deferred to the Appendices B.4.1 and B.4.2.

As shown in Tables 2 and 3, DEPT demon-

Table 3: Out-of-distribution on reasoning benchmarks. The best results are **bolded**. The second results are underline.

Methods	Minerva	AMC-23	AIME24	AIME25	Math500	Olympiad	GPQA-D	MMLU-Pro	Average
<i>Qwen3-4B-Base</i>									
VANILLA	22.36	42.03	9.58	6.88	65.80	34.52	28.79	39.36	31.07
SPAG	24.70	44.27	<u>10.69</u>	7.78	<u>70.87</u>	<u>36.15</u>	33.33	52.71	35.06
GRPO	23.25	38.65	7.43	5.80	56.26	26.64	<u>35.86</u>	49.14	30.38
MARS	<u>29.57</u>	45.47	9.41	<u>8.47</u>	69.25	35.18	34.01	<u>53.34</u>	<u>35.59</u>
SPIRAL	<u>27.17</u>	<u>45.73</u>	9.55	7.81	67.95	34.15	34.18	51.32	34.73
DEPT (Ours)	32.57	48.70	11.22	10.03	74.64	38.79	37.04	56.45	38.68
<i>Qwen3-8B-Base</i>									
VANILLA	24.15	47.66	13.44	10.00	67.54	35.80	29.80	46.97	34.42
SPAG	26.28	50.42	12.78	10.76	69.67	37.21	31.14	51.46	36.22
GRPO	<u>34.24</u>	<u>53.75</u>	12.50	<u>12.05</u>	75.94	<u>40.56</u>	35.69	59.05	<u>40.47</u>
MARS	<u>33.75</u>	<u>53.59</u>	14.03	11.74	74.63	40.35	<u>37.37</u>	57.80	40.41
SPIRAL	31.71	53.59	<u>13.54</u>	10.38	72.81	38.40	35.69	<u>58.28</u>	39.30
DEPT (Ours)	35.00	57.03	13.06	12.43	<u>74.98</u>	40.83	38.72	57.60	41.21

strates superior out-of-distribution generalization. In out-of-distribution games, our approach maintains robust performance even in more complex environments. For QA and mathematics, self-play on games enables effective reasoning ability transfer to general reasoning tasks. This aligns with prior findings (Cheng et al., 2024; Liu et al., 2025a; Yuan et al., 2025) that game-based self-play effectively elicits reasoning capabilities—since DEPT achieves stronger in-game performance with more strategic behaviors, it consequently yields greater improvements on reasoning benchmarks. The improvements are particularly pronounced on Qwen3-4B-Base, where consistent gains are observed across all benchmarks.

RQ5: Additional results. We provide several supplementary analyses in the appendix to further validate the effectiveness of DEPT. First, we examine the semantic diversity of generated actions to verify whether DEPT can produce more varied and contextually appropriate behaviors rather than repetitive patterns. The results in Figure 8 demonstrate that DEPT achieves higher semantic richness compared to baselines, indicating that our dual-process mechanism encourages more creative and diverse action generation (See Appendix C). Second, we conduct a sensitivity analysis on the key hyperparameter α_{fast} , which controls the balance between fast and slow thinking modes. Our analysis in Table 5 reveals that DEPT maintains stable performance across the range [0.4, 0.6], suggesting that our method is robust to hyperparameter choices and does not require extensive tuning for deployment (See Appendix D). Third, we analyze

the computational overhead introduced by our dual-process architecture to assess its practical applicability. The results in Figure 9 show that DEPT introduces negligible additional cost (less than 0.0016% of the total per-iteration training time) while achieving substantial performance improvements, making it suitable for real-world applications where efficiency is critical (See Appendix E). Additionally, we perform comprehensive statistical significance tests to rigorously validate our experimental findings. The results in Table 6 confirm that DEPT provides statistically significant gains over all baselines ($p < 0.05$), demonstrating that the observed improvements are not due to random variance (See Appendix F).

6 Related Work

LLM-based Reinforcement Learning Reinforcement learning for LLMs has evolved from Reinforcement Learning from Human Feedback (RLHF) to RLVR (Ouyang et al., 2022; Guo et al., 2025; Team et al., 2025; Yue et al., 2025; Liu et al., 2025b; Tan et al., 2025; Nie et al., 2026; Tan et al., 2026), leveraging objective ground truth to unlock deep potential in mathematical reasoning and code generation. However, existing RLVR research focuses on closed-ended tasks with well-defined answers. Real-world open-ended social interactions—such as negotiation and deception—are fundamentally different (FAIR et al., 2022; Park et al., 2023; Zhou et al., 2024; Xu et al., 2024; Wang et al., 2025b; Zhang et al., 2026): they involve infinite action spaces, lack unique correct strategies, and require real-time reasoning about dynamic oppo-

intentions (Abdulhai et al., 2025; Yang et al., 2025b; Wang et al., 2025a). Our work addresses the challenge of applying RLVR to open-ended social interaction while sustaining continuous evolution in dynamic environments.

LLM Self-play Training on Language Game

As a classical paradigm, Self-play (Silver et al., 2018; Bansal et al., 2018; Chen et al., 2024; Cheng et al., 2024) addresses data scarcity while constructing an automatic curriculum through opponent co-evolution. For language games in LLMs, SPIRAL (Liu et al., 2025a) and MARS (Yuan et al., 2025) introduce RAE to handle role asymmetry, achieving strong performance in limited action spaces. However, in more complex open-ended language domains, these methods suffer from evolutionary impasse: LLMs converge to homogeneous behaviors, collapsing outcome distributions into deterministic results. This drives advantages toward zero, causing gradient vanishing and trapping models in suboptimal states. Our DEPT addresses this through dual-baseline perception to detect stagnation, combined with asymmetric advantage reshaping to restore optimization landscape and ensure continuous policy evolution.

7 Conclusions

In this paper, we address the critical challenge of evolution impasse in applying self-play RLVR to open-ended social language games, where agent training is susceptible to gradient vanishing and strategic stagnation. We propose DEPT, which dynamically perceives evolutionary states and adaptively adjusts the optimization landscape via dual-timescale value baselines. By quantifying stagnation through baseline divergence and match entropy, DEPT selectively intervenes to penalize dominant outcomes while amplifying rare trajectories, effectively restoring gradient signals and enforcing continuous strategic exploration. Extensive empirical results confirm that our approach not only surpasses SOTA methods but also maintains superior out-of-distribution generalization capabilities.

Limitations

While DEPT demonstrates substantial improvements in preventing evolutionary impasse during self-play training, several limitations offer avenues for future research. (1) Self-play reinforcement learning inherently demands computational resources. Our experiments required approximately

30 GPU-hours per run on 8 NVIDIA A800 GPUs. These resource requirements may limit accessibility for researchers with constrained computational budgets. (2) Following the reasoning-then-acting paradigm, our method generates extended response sequences containing both reasoning traces (c_t) and executable actions (a_t). This increases inference latency compared to direct action generation, a common trade-off in test-time scaling approaches. The computational cost becomes more pronounced when scaling to longer interaction horizons or larger LLM backbones. We believe our dual-scale evolutionary policy training method provides a principled foundation that could inspire novel mechanisms for detecting and intervening in more complex evolutionary dynamics, and we hope this work motivates the community to explore these promising directions.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grants #72293575, #72225011 and #72434005. We also thank Longze Chen and Jiaming Li for their valuable suggestions on the experimental design of this paper.

References

- Marwa Abdulhai, Isadora White, Charlie Victor Snell, Charles Sun, Joey Hong, Yuexiang Zhai, Kelvin Xu, and Sergey Levine. 2025. *LMRL gym: Benchmarks for multi-turn reinforcement learning with language models*. In *Proceedings of ICML*.
- Alexh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. 2019. Reinforcement learning: Theory and algorithms. <https://rltheorybook.github.io/>.
- Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. 2018. *Emergent complexity via multi-agent competition*. In *Proceedings of ICLR*.
- Federico Bianchi, Patrick John Chia, Mert Yuksekgonul, Jacopo Tagliabue, Dan Jurafsky, and James Zou. 2024. *How well can llms negotiate? negotiation-arena platform and analysis*. In *Proceedings of ICML*, pages 3935–3951.
- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. *Self-play fine-tuning converts weak language models to strong language models*. In *Proceedings of ICML*, pages 6621–6642.
- Pengyu Cheng, Tianhao Hu, Han Xu, Zhisong Zhang, Yong Dai, Lei Han, Nan Du, and Xiaolong Li. 2024. *Self-playing adversarial language game enhances llm*

- reasoning. In *Proceedings of NeurIPS*, pages 126515–126543.
- Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. 2024. *Gtbench: Uncovering the strategic reasoning capabilities of llms via game-theoretic evaluations*. *Proceedings of NeurIPS*, 37:28219–28253.
- Meta FAIR, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, and 1 others. 2022. *Human-level play in the game of diplomacy by combining language models with strategic reasoning*. *Science*, 378(6624):1067–1074.
- Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. 2025. *Textarena*. *arXiv preprint arXiv:2504.11442*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. *Deepseek-r1 incentivizes reasoning in llms through reinforcement learning*. *Nature*, 645(8081):633–638.
- Thilo Hagendorff. 2023. *Deception abilities emerged in large language models*. *arXiv preprint arXiv:2307.16513*.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. *OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems*. In *Proceedings of ACL*, pages 3828–3850.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. *Measuring mathematical problem solving with the MATH dataset*. In *Proceedings of NeurIPS Datasets and Benchmarks Track*.
- Wenyue Hua, Ollie Liu, Lingyao Li, Alfonso Amayuelas, Julie Chen, Lucas Jiang, Mingyu Jin, Lizhou Fan, Fei Sun, William Wang, and 1 others. 2024. *Game-theoretic llm: Agent workflow for negotiation games*. *arXiv preprint arXiv:2411.05990*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. *Efficient memory management for large language model serving with pagedattention*. In *Proceedings of the SIGOPS*.
- John E Laird. 2019. *The Soar cognitive architecture*. MIT press.
- Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. *Solving quantitative reasoning problems with language models*. In *Proceedings of NeurIPS*, volume 35, pages 3843–3857.
- Michael L Littman. 1994. *Markov games as a framework for multi-agent reinforcement learning*. In *Machine learning proceedings 1994*, pages 157–163. Elsevier.
- Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, and 1 others. 2025a. *Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning*. *arXiv preprint arXiv:2506.24119*.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025b. *Understanding r1-zero-like training: A critical perspective*. In *Proceedings of COLM*.
- Zichen Liu, Changyu Chen, Xinyi Wan, Chao Du, Wee Sun Lee, and Min Lin. 2024. *Oat: A research-friendly framework for llm online alignment*. <https://github.com/sail-sg/oat>.
- MAA. 2023. *American mathematics contest 12 (amc 12)*.
- MAA. 2024. *American invitational mathematics examination (aime)*.
- MAA. 2025. *American invitational mathematics examination (aime)*.
- Shuaiyi Nie, Siyu Ding, Wenyuan Zhang, Linhao Yu, Tianmeng Yang, Yao Chen, Tingwen Liu, Weichong Yin, Yu Sun, and Hua Wu. 2026. *Attnpo: Attention-guided process supervision for efficient reasoning*. *arXiv preprint arXiv:2602.09953*.
- OpenAI. 2024. *simple-evals: A suite of lightweight evaluations for language models*. <https://github.com/openai/simple-evals>.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. *Training language models to follow instructions with human feedback*. In *Proceedings of NeurIPS*, volume 35, pages 27730–27744.
- Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirota, and Marcello Restelli. 2018. *Stochastic variance-reduced policy gradient*. In *Proceedings of ICML*, volume 80, pages 4026–4035.
- Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. *Generative agents: Interactive simulacra of human behavior*. In *Proceedings of UIST*, pages 1–22.

- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. [GPQA: A graduate-level google-proof q&a benchmark](#). In *Proceedings of COLM*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *arXiv preprint arXiv:2402.03300*.
- David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, and 1 others. 2018. [A general reinforcement learning algorithm that masters chess, shogi, and go through self-play](#). *Science*, 362(6419):1140–1144.
- Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas Griffiths. 2023. [Cognitive architectures for language agents](#). *Transactions on Machine Learning Research*.
- Yuqiao Tan, Minzheng Wang, Shizhu He, Huanxuan Liao, Chengfeng Zhao, Qiunan Lu, Tian Liang, Jun Zhao, and Kang Liu. 2025. [Bottom-up policy optimization: Your language model policy secretly contains internal policies](#). *arXiv preprint arXiv:2512.19673*.
- Yuqiao Tan, Minzheng Wang, Bo Liu, Zichen Liu, Tian Liang, Shizhu He, Jun Zhao, and Kang Liu. 2026. [From \$p\(y|x\)\$ to \$p\(y\)\$: Investigating reinforcement learning in pre-train space](#).
- Kimi Team, Angang Du, Bofei Gao, Bawei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. [Kimi k1. 5: Scaling reinforcement learning with llms](#). *arXiv preprint arXiv:2501.12599*.
- Minzheng Wang, Yongbin Li, Haobo Wang, Xinghua Zhang, Nan Xu, Bingli Wu, Fei Huang, Haiyang Yu, and Wenji Mao. 2025a. [Adaptive thinking via mode policy optimization for social language agents](#). *arXiv preprint arXiv:2505.02156*.
- Minzheng Wang, Xinghua Zhang, Kun Chen, Nan Xu, Haiyang Yu, Fei Huang, Wenji Mao, and Yongbin Li. 2025b. [DEMO: Reframing dialogue interaction with fine-grained element modeling](#). In *Findings of ACL*, pages 11373–11401.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. 2024. [Mmlu-pro: A more robust and challenging multi-task language understanding benchmark](#). In *Proceedings of NeurIPS*, volume 37, pages 95266–95290. Curran Associates, Inc.
- Tian Xia, Zhiwei He, Tong Ren, Yibo Miao, Zhuosheng Zhang, Yang Yang, and Rui Wang. 2024. [Measuring bargaining abilities of llms: A benchmark and a buyer-enhancement method](#). In *Findings of ACL*, pages 3579–3602.
- Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. 2024. [Language agents with reinforcement learning for strategic play in the werewolf game](#). In *Proceedings of ICML*, pages 55434–55464.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. [Qwen3 technical report](#). *arXiv preprint arXiv:2505.09388*.
- Ruihan Yang, Yikai Zhang, Aili Chen, Xintao Wang, Jiangjie Chen, Siyu Yuan, Deqing Yang, and Yanghua Xiao. 2025b. [ARIA: Training language agents with intention-driven reward aggregation](#). In *Proceedings of NeurIPS*.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. [React: Synergizing reasoning and acting in language models](#). In *Proceedings of ICLR*.
- Huining Yuan, Zelai Xu, Zheyue Tan, Xiangmin Yi, Mo Guang, Kaiwen Long, Haojia Hui, Boxun Li, Xinlei Chen, Bo Zhao, and 1 others. 2025. [Mars: Reinforcing multi-agent reasoning of llms through self-play in strategic games](#). *arXiv preprint arXiv:2510.15414*.
- Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. [Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model?](#) In *Proceedings of NeurIPS*.
- Wenyuan Zhang, Xinghua Zhang, Haiyang Yu, Shuaiyi Nie, Bingli Wu, Juwei Yue, Tingwen Liu, and Yongbin Li. 2026. [Expseek: Self-triggered experience seeking for web agents](#). *arXiv preprint arXiv:2601.08605*.
- Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. 2024. [SOTOPIA: Interactive evaluation for social intelligence in language agents](#). In *Proceedings of ICLR*.

A Theoretical Analysis

In this section, we provide a theoretical analysis of DEPT. We first establish the mathematical foundation of the Time-scaled Evolutionary Perception mechanism, proving that the proposed metrics serve as estimators for the velocity and variance of the reward distribution. Subsequently, we analyze

the gradient dynamics, proving that our method ensures effectively restoring gradient signals in evolutionary impasse via a synthetic variance injection mechanism.

A.1 Mathematical Foundation of Evolutionary Perception

The core of DEPT lies in quantifying the evolutionary state using baseline divergence and match entropy. Here, we provide the formal derivations linking these metrics to the underlying reward dynamics.

A.1.1 Baseline Divergence as Velocity Estimation

In the main text, we utilize the divergence between fast and slow baselines to measure training stationarity. We now prove that this metric is mathematically equivalent to an estimator of the *velocity of expected return*.

Proposition A.1 (Divergence-Velocity Equivalence). *Let $\mu(t) = \mathbb{E}[R_p(\tau)]$ be the underlying expected return at step t . Assuming $\mu(t)$ is locally linear within the effective window of the Exponential Moving Average (EMA), the expected divergence between two baselines with decay rates $\alpha_{slow} > \alpha_{fast}$ is proportional to the magnitude of the time derivative of the expected return:*

$$\mathbb{E} [|b_p^{fast,t} - b_p^{slow,t}|] \approx C \cdot \left| \frac{d\mu(t)}{dt} \right|, \quad (10)$$

where $C > 0$ is a constant determined by the time constants.

Proof. Recall the EMA update rule: $b_p^{k,t} = \alpha_k b_p^{k,t-1} + (1 - \alpha_k) R_p(\tau)$. By recursively expanding this update, the baseline at time t can be expressed as a weighted sum of historical rewards. Taking the expectation and performing a first-order Taylor expansion of the expected return $\mu(t - j) \approx \mu(t) - j \cdot \dot{\mu}(t)$:

$$\begin{aligned} \mathbb{E}[b_p^{k,t}] &= (1 - \alpha_k) \sum_{j=0}^{\infty} \alpha_k^j \mu(t - j) \\ &\approx (1 - \alpha_k) \sum_{j=0}^{\infty} \alpha_k^j (\mu(t) - j \cdot \dot{\mu}(t)) \\ &= \underbrace{\mu(t) (1 - \alpha_k) \sum_{j=0}^{\infty} \alpha_k^j}_{=1} - \dot{\mu}(t) (1 - \alpha_k) \sum_{j=0}^{\infty} j \alpha_k^j. \end{aligned} \quad (11)$$

Using the geometric series identity $\sum_{j=0}^{\infty} j x^j = \frac{x}{(1-x)^2}$ for $|x| < 1$, we substitute $x = \alpha_k$:

$$\mathbb{E}[b_p^{k,t}] \approx \mu(t) - \dot{\mu}(t) \cdot \frac{\alpha_k}{1 - \alpha_k}. \quad (12)$$

Let $\mathcal{T}_k = \frac{\alpha_k}{1 - \alpha_k}$ denote the effective time lag. The expected divergence is:

$$\begin{aligned} \mathbb{E}[\delta^{(t)}] &= |\mathbb{E}[b_p^{fast,t}] - \mathbb{E}[b_p^{slow,t}]| \\ &\approx |(\mu(t) - \mathcal{T}_{fast} \dot{\mu}(t)) - (\mu(t) - \mathcal{T}_{slow} \dot{\mu}(t))| \\ &= (\mathcal{T}_{slow} - \mathcal{T}_{fast}) \cdot |\dot{\mu}(t)|. \end{aligned} \quad (13)$$

Since $\alpha_{slow} > \alpha_{fast}$, we have $\mathcal{T}_{slow} > \mathcal{T}_{fast}$, ensuring the coefficient is positive. Thus, the baseline divergence serves as a direct proxy for the training velocity $|\dot{\mu}(t)|$.

A.1.2 Match Entropy as Time-Varying Variance Proxy

We now establish the mathematical relationship between the observed Match Entropy $H_{match}^{(t)}$ and the underlying reward variance.

Lemma A.1 (Entropy-Variance Correlation). *Let $\nu(t) = \text{Var}[R_p(\tau)]$ be the underlying reward variance at time step t . As the match entropy vanishes, the reward variance asymptotically approaches zero:*

$$\lim_{H_{match}^{(t)} \rightarrow 0} \nu(t) = 0. \quad (14)$$

Consequently, maintaining high entropy is a necessary condition for maintaining non-zero reward variance.

Proof. Consider the discrete outcome space $\mathcal{O} = \{o_1, \dots, o_K\}$ (e.g., {Win, Loss, Draw}) with associated fixed reward values $\{r_1, \dots, r_K\}$. At time t , the policy π_{θ_t} induces a probability distribution $p_i(t) = P(R_p(\tau) = r_i)$.

The variance $\nu(t)$ is formally defined as:

$$\nu(t) = \sum_{i=1}^K p_i(t) (r_i - \mathbb{E}[R_p(\tau)])^2. \quad (15)$$

From information theory, the entropy $H_{match}^{(t)} = -\sum p_i(t) \log p_i(t)$ approaches zero if and only if the probability mass concentrates on a single dominant outcome o_k . Mathematically, this implies the limit behavior:

$$p_k(t) \rightarrow 1 \quad \text{and} \quad p_{j \neq k}(t) \rightarrow 0. \quad (16)$$

In this limit, the expected return converges to the reward of the dominant outcome: $\mathbb{E}[R_p(\tau)] \rightarrow r_k$. Substituting these limits into the definition of $\nu(t)$:

$$\begin{aligned} \lim_{H_{\text{match}}^{(t)} \rightarrow 0} \nu(t) &= \lim_{p_k \rightarrow 1} \left[p_k(t)(r_k - r_k)^2 \right. \\ &\quad \left. + \sum_{j \neq k} p_j(t)(r_j - r_k)^2 \right] \quad (17) \\ &= 1 \cdot 0 + 0 = 0. \end{aligned}$$

This derivation proves that $H_{\text{match}}^{(t)}$ and $\nu(t)$ are positively correlated in the asymptotic limit: a collapse in match entropy strictly necessitates a collapse in the reward variance $\nu(t)$.

A.1.3 Bayesian Derivation of Intervention Coefficient

We presented the intervention coefficient $\lambda^{(t)}$ in a specific functional form in the main text. Here, we derive this form from a general Bayesian perspective.

Theorem A.1 (Bayesian Form of λ). *Let $Z_t \in \{0, 1\}$ be a latent variable where $Z_t = 1$ indicates a evolution impasse. The coefficient $\lambda^{(t)}$ represents the posterior probability $P(Z_t = 1 \mid \mathcal{O}_t)$ given observations \mathcal{O}_t .*

Proof. We decompose the degenerate state into two independent necessary conditions: Stationarity (\mathcal{S}) and Determinism (\mathcal{D}). Using Bayes' theorem and assuming conditional independence:

$$\lambda^{(t)} \triangleq P(Z_t = 1) \propto P(\mathcal{S} \mid \delta^{(t)}) \cdot P(\mathcal{D} \mid H_{\text{match}}^{(t)}). \quad (18)$$

To map observations to probabilities, we introduce two kernel functions $\mathcal{K}_{\mathcal{S}}$ and $\mathcal{K}_{\mathcal{D}}$ with range $[0, 1]$:

1. **Stagnation Kernel $\mathcal{K}_{\mathcal{S}}$:** Since $\delta^{(t)} \propto |\text{Velocity}|$, stationarity implies $\delta^{(t)} \rightarrow 0$. We select the hyperbolic tangent kernel:

$$\sigma^{(t)} = P(\mathcal{S} \mid \delta^{(t)}) = 1 - \tanh(\delta^{(t)}). \quad (19)$$

2. **Determinism Kernel $\mathcal{K}_{\mathcal{D}}$:** Since low entropy implies determinism, we select the radical kernel to ensure high sensitivity near the boundary of exploration ($H \approx 1$):

$$\gamma^{(t)} = P(\mathcal{D} \mid H_{\text{match}}^{(t)}) = \sqrt{1 - H_{\text{match}}^{(t)}}. \quad (20)$$

Substituting these specific kernels yields the instantiation used in DEPT: $\lambda^{(t)} = \sigma^{(t)} \cdot \gamma^{(t)}$.

A.2 Gradient Dynamics Analysis

A.2.1 The Vanishing Gradient of Standard Methods

We first prove why standard methods fail in evolutionary impasse.

Theorem A.2 (Variance-Dependent Gradient Bound). *Assume the policy π_{θ} is Lipschitz continuous such that the norm of the score function is bounded by a constant G_{max} . The magnitude of the expected policy gradient in standard Advantage Estimation is upper-bounded by the standard deviation of the reward:*

$$\|\nabla_{\theta} J(\theta)\| \leq G_{\text{max}} \cdot \sqrt{\nu(t)}. \quad (21)$$

Proof. We start with the standard gradient definition $\nabla_{\theta} J = \mathbb{E}_{\tau}[(R - \mu)\nabla_{\theta} \log \pi_{\theta}]$. First, we invoke the **Lipschitz continuity assumption** (Papini et al., 2018; Agarwal et al., 2019): for smooth neural policies, the gradient of the log-likelihood is bounded:

$$\sup_{\tau} \|\nabla_{\theta} \log \pi_{\theta}(\tau)\| \leq G_{\text{max}}. \quad (22)$$

Next, we apply the **Cauchy-Schwarz inequality** for random variables ($\mathbb{E}[XY]^2 \leq \mathbb{E}[X^2]\mathbb{E}[Y^2]$):

$$\begin{aligned} \|\nabla_{\theta} J\|^2 &= \|\mathbb{E}[(R - \mu)\nabla \log \pi]\|^2 \\ &\leq \mathbb{E}[(R - \mu)^2] \cdot \mathbb{E}[\|\nabla \log \pi\|^2] \quad (23) \\ &= \nu(t) \cdot \mathbb{E}[\|\nabla \log \pi\|^2]. \end{aligned}$$

Substituting the bound G_{max} :

$$\|\nabla_{\theta} J\|^2 \leq \nu(t) \cdot G_{\text{max}}^2 \implies \|\nabla_{\theta} J\| \leq G_{\text{max}} \sqrt{\nu(t)}. \quad (24)$$

Implication: In a evolution impasse, Lemma A.1.2 dictates that $\nu(t) \rightarrow 0$. Consequently, the gradient magnitude $\|\nabla J\|$ is forced to zero. The optimization halts specifically because the *signal variance* $\nu(t)$ required to drive the gradient has vanished.

A.2.2 Gradient Restoration

We prove that DEPT can effectively recovers gradient.

Theorem A.3 (Push-Pull Gradient Restoration). *In a degenerate state where natural variance $\nu(t) \rightarrow 0$, DEPT constructs a non-vanishing synthetic gradient field driven by the historical value range $[V_{\text{min}}, V_{\text{max}}]$.*

Proof. Consider a training batch \mathcal{B} of size B , decomposed into a set of dominant trajectories \mathcal{D}_{dom}

(size M) and rare trajectories $\mathcal{D}_{\text{rare}}$ (size N), such that $B = M + N$ and $M \gg N$ (indicating a degenerate state). When intervention is triggered ($\lambda^{(t)} \rightarrow 1$), the gradient update ∇J_p is calculated as the summation over these two subsets:

$$\nabla J_p \approx \frac{1}{B} \left[\underbrace{\sum_{\tau \in \mathcal{D}_{\text{dom}}} (R_p(\tau) - V_{\text{max}}) \nabla \log \pi_p(\tau)}_{\text{Aggregate Push Force}} + \underbrace{\sum_{\tau' \in \mathcal{D}_{\text{rare}}} (R_p(\tau') - V_{\text{min}}) \nabla \log \pi_p(\tau')}_{\text{Aggregate Pull Force}} \right]. \quad (25)$$

1. **Symmetric Dilemma:** In standard methods, the baseline $b \approx \frac{1}{M} \sum R_p(\tau_{\text{dom}})$. Thus, the advantage for the M dominant samples approaches zero. Since $N \rightarrow 0$, the total gradient vanishes.
2. **Aggregate Push Mechanism (M samples):** For the dominant set, we set the baseline to V_{max} . Since $V_{\text{max}} \geq R_p(\tau)$ for $\tau \in \mathcal{D}_{\text{dom}}$, the term $(R_p(\tau) - V_{\text{max}})$ is strictly non-positive. Although individual gradients may be small, the summation over M samples generates a substantial negative gradient field, actively suppressing the probability of the stagnant strategy.
3. **Aggregate Pull Mechanism (N samples):** For the rare set, we set the baseline to V_{min} . The term $(R_p(\tau') - V_{\text{min}})$ is maximized, assigning a high positive weight to these sparse signals. This amplifies the exploration direction even when N is small.

Through this summation, DEPT effectively injects a **Synthetic Variance** $\nu_{\text{syn}} \propto (V_{\text{max}} - V_{\text{min}})^2$ into the batch. Unlike the natural variance $\nu(t)$ which vanishes as $N \rightarrow 0$, this synthetic variance remains robust, ensuring $\|\nabla J_p\| > 0$ and forcing the policy distribution to shift away from the evolution impasse.

B Implementation details

B.1 Baselines

For a fair comparison, all methods use the same amount of training transitions for each task. For online RL methods (e.g., SPIRAL, MARS, GRPO),

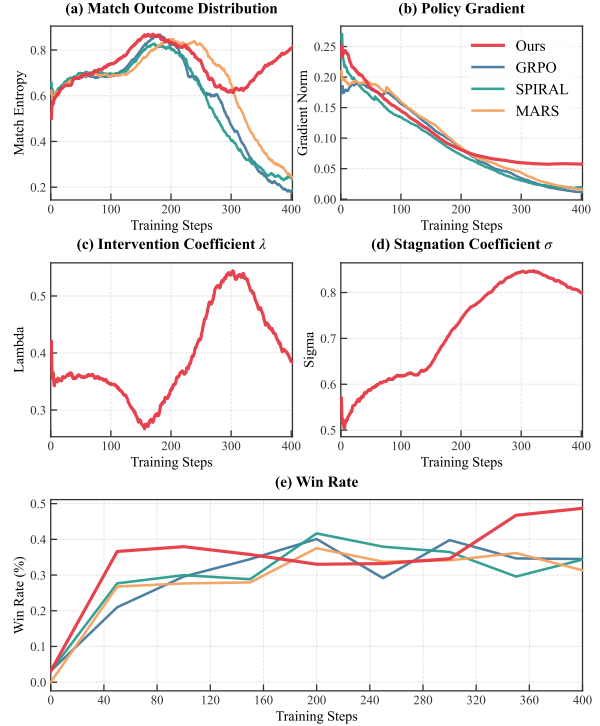


Figure 6: Training dynamics on Don't Say It.

all RL hyperparameters and configurations remain consistent with our method, except for their respective advantage estimation schemes. For the offline RL method (e.g., SPAG), we adopt the same reward computation logic and three-iteration update procedure as in the original work. Due to significant performance degradation of baseline methods on the Negotiation, we report their best checkpoints selected based on training curves, while using the final checkpoint (step 400) for our method; for other tasks (Don't Say It, Two Dollar), all methods are evaluated at the final checkpoint (step 400).

B.2 Tasks

Don't Say It tests linguistic manipulation and theory of mind through adversarial dialogue. Each player is assigned a secret target word and must guide their opponent into saying it during natural conversation, while avoiding their opponent's unknown target word. Success depends on tracking dialogue states, predicting opponent responses, and detecting potential traps. Beyond strategic reasoning, players should maintain natural conversational flow to avoid arousing suspicion. These capabilities are relevant to persuasion, deception detection, and defense against social engineering. The environment ID in TextArena is DontSayIt-v0 with `max_turns=20`.

Algorithm 1 DEPT: Dual-scale Evolutionary Policy Training

Require: Policy π_θ , decay rates $\alpha_{\text{fast}} < \alpha_{\text{slow}}$

```
1: Initialize baselines  $b_p^{\text{fast}} = b_p^{\text{slow}} = 0$  for  $p \in \{0, 1\}$ 
2: while not converged do
3:   // Self-Play Trajectory Collection
4:   Collect batch  $\mathcal{B} = \{\tau_k\}_{k=1}^K$  via self-play, where each  $\tau_k$  contains alternating
      actions  $\{y_t^{(p)}\}$  from  $\pi_\theta(\cdot | o_t, p)$  and terminal rewards  $R_0 = -R_1 = \rho(s_T)$ 
5:   // Batch-Level Statistics
6:   Compute outcome distribution  $P(o | p)$  for  $o \in \{\text{win, draw, loss}\}$  from  $\mathcal{B}$ 
7:   Identify dominant outcome  $o_{\text{dom}}^p \leftarrow \arg \max_o P(o | p)$  for each  $p$ 
8:   Compute normalized match entropy  $H_{\text{match}} \leftarrow -\frac{1}{\log 3} \sum_o P(o) \log P(o)$ 
9:   // Dual-Timescale Update & Asymmetric Advantage Reshaping
10:  for  $\tau \in \mathcal{B}$  do
11:    for  $p \in \{0, 1\}$  do
12:      // Update dual-timescale baselines and global bounds
13:       $b_p^{\text{fast}} \leftarrow \alpha_{\text{fast}} \cdot b_p^{\text{fast}} + (1 - \alpha_{\text{fast}}) \cdot R_p(\tau)$ 
14:       $b_p^{\text{slow}} \leftarrow \alpha_{\text{slow}} \cdot b_p^{\text{slow}} + (1 - \alpha_{\text{slow}}) \cdot R_p(\tau)$ 
15:       $V_{\text{max}} \leftarrow \max(V_{\text{max}}, b_p^{\text{fast}})$ ,  $V_{\text{min}} \leftarrow \min(V_{\text{min}}, b_p^{\text{fast}})$ 
16:      // Compute intervention coefficient via baseline divergence
17:       $\sigma_p \leftarrow 1 - \tanh\left(\frac{|b_p^{\text{fast}} - b_p^{\text{slow}}|}{H_{\text{match}}}\right)$  ▷ Stagnation coefficient
18:       $\lambda_p \leftarrow \sigma_p \cdot \sqrt{1 - H_{\text{match}}}$  ▷ Intervention coefficient
19:      // Construct asymmetric baseline via outcome scarcity
20:      if  $o_\tau = o_{\text{dom}}^p$  then
21:         $V_{\text{asym}}(\tau) \leftarrow V_{\text{max}}$ 
22:      else
23:         $V_{\text{asym}}(\tau) \leftarrow V_{\text{min}}$ 
24:      end if
25:      // Fuse baselines and compute reshaped advantage
26:       $\tilde{b}_p(\tau) \leftarrow (1 - \lambda_p) \cdot b_p^{\text{slow}} + \lambda_p \cdot V_{\text{asym}}(\tau)$ 
27:       $\tilde{A}_p(\tau) \leftarrow R_p(\tau) - \tilde{b}_p(\tau)$ 
28:    end for
29:  end for
30:  Update  $\theta$  using policy gradient with reshaped advantages  $\tilde{A}_p$ 
31: end while
```

Negotiation involves multi-attribute trading under incomplete information. Players trade five commodity types (Wheat, Wood, Sheep, Brick, Ore) to maximize their portfolio value, determined by private utility functions unknown to the opponent. The action space supports bundled offers, rejections, and binding acceptances. Success requires not only inferring opponent preferences and identifying mutually beneficial trades, but also articulating proposals persuasively and building cooperative rapport through dialogue. These skills transfer to resource allocation and mixed-motive multi-agent coordination. The environment ID in TextArena is SimpleNegotiation-v0 with max_turns=10. Note that our setup matches the original game spec-

ification; in contrast, SPIRAL simplified the environment from five commodities to two.

Two Dollar models bargaining with hidden constraints. Players divide a fixed pool (e.g., \$2.00) while following secret role-specific instructions that impose behavioral restrictions (e.g., aggression levels) or minimum acceptable amounts. Success requires inferring opponent constraints from behavioral cues, adapting communication style to different personas, and using persuasive dialogue to reach agreement while maintaining role consistency. These capabilities are relevant to constrained decision-making and human-AI coordination in mixed-motive settings. The environment ID in TextArena is TwoDollar-v0 with max_turns=10.

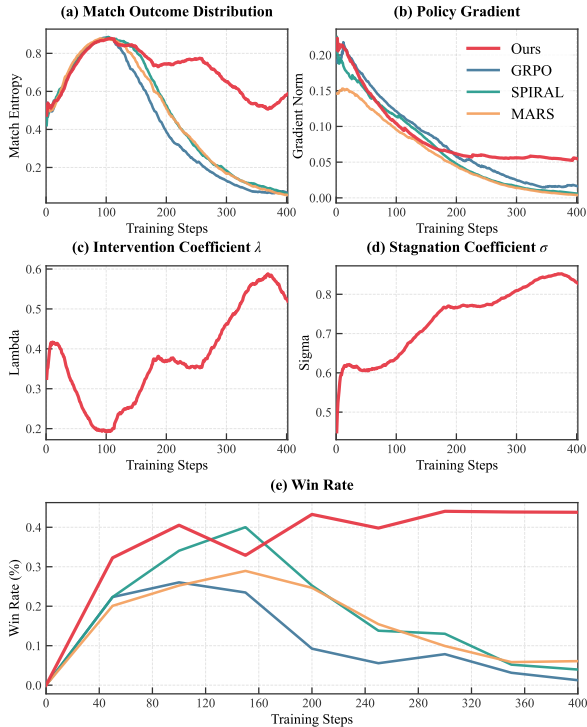


Figure 7: Training dynamics on Negotiation.

Across all three tasks, we align our reward mechanism with the SPIRAL (Liu et al., 2025a): a win yields a reward of +1, a loss yields -1 , and a draw yields 0. Additionally, format errors incur a penalty of -1.5 .

B.3 Training Hyperparameters

These hyperparameters in Table 4 are held constant across all game environments and model scales to ensure a fair comparison. Our implementation leverages vLLM (Kwon et al., 2023) and Oat (Liu et al., 2024), establishing a robust distributed actor-learner architecture. The training infrastructure is deployed on 8 NVIDIA A800 GPUs, where parallel actors collect game trajectories concurrently while a centralized learner executes synchronous policy updates. Each experimental run required approximately 30 GPU-hours on a 8 NVIDIA A800 GPUs server.

B.4 Out-of-distribution Evaluation

B.4.1 Increased Complexity Games

HardCore Don’t Say It increases difficulty by shifting from common words to open-domain vocabulary. By enabling *hardcore* mode, target words are sampled from the full English lexicon, including abstract, low-frequency, and polysemous terms, rather than a simplified basic vocabulary. The con-

Parameter	Value
Maximum response length	4096 tokens
Sampling temperature	1.0
(top P, top K)	(1.0, -1)
Optimizer	AdamW
Adam parameters (β_1, β_2)	(0.9, 0.95)
Weight decay	0.0
Gradient norm clipping	1.0
Batch size	128
Training steps	400
Discount factor	0.9
EMA decay rate	0.95
Learning rate scheduler	Constant
Learning rate	1×10^{-6}
Inner proximal update epoch	1
KL loss coefficient	0.0
KL penalty coefficient	0.0
Policy clipping parameter	0.2

Table 4: Hyperparameter configurations used in all experiments.

versation horizon is also reduced from 20 to 10 turns. This setup tests efficient semantic planning and generalization, requiring agents to rely on flexible reasoning rather than memorized steering patterns.

Random Value Negotiation evaluates adaptation to non-stationary preferences. While the standard version follows a fixed utility ranking (e.g., Ore $>$ Wheat), the OOD setting randomizes resource valuations at each episode. Combined with a reduced horizon of 6 turns, agents cannot rely on learned priors about resource value. Success requires rapid inference of the opponent’s preferences and effective trade formulation under time pressure.

Variable Sum Dollar tests numerical generalization and strict constraint adherence. Unlike the standard fixed-sum setting (\$2.00), this variant randomizes the total pool size (ranging from \$2 to \$20), requiring agents to generalize their division strategies to unseen scales. The difficulty is increased by a shortened horizon of 6 rounds. This environment assesses whether agents can maintain logical consistency and role compliance when facing novel numerical parameters.

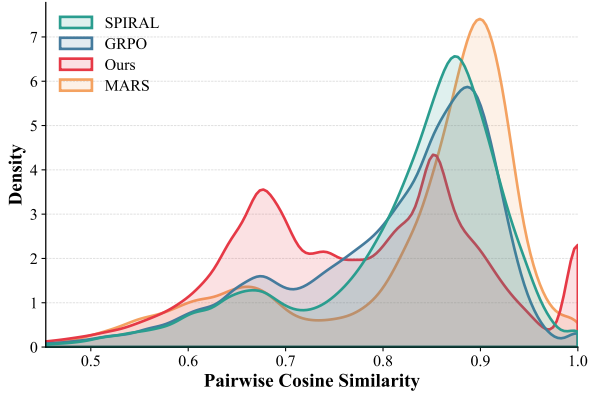


Figure 8: Semantic diversity of actions.

B.4.2 General Reasoning Benchmark

To assess the transferability of reasoning skills from gameplay to general domains, we evaluate our models on a suite of established benchmarks in the zero-shot evaluation setting. For each RL algorithm and backbone LLM, we report the averaged performance of three models trained on distinct social language games. All inference is performed with a temperature of 0.6 and a top-p of 0.95.

Math Benchmark We report AVG@32 for AIME24 (MAA, 2024), AIME25 (MAA, 2025) and AVG@16 for Minerva (Lewkowycz et al., 2022), AMC-23 (MAA, 2023), Math500 (Hendrycks et al., 2021), Olympiad (He et al., 2024). These benchmark cover a wide range of topics including algebra, geometry, and competitive mathematic.

Question Answering Benchmark We utilize GPQA-Diamond (Rein et al., 2024), which consists of graduate-level science questions and MMLU-Pro (Wang et al., 2024), a benchmark for multidisciplinary knowledge. We use simple-evals (OpenAI, 2024) for evaluation.

C Semantic Diversity Analysis

To further analyze the effectiveness of our method, we evaluate the semantic diversity of generated actions from the perspective of semantic space. Specifically, we employ an embedding model (Qwen3-Embedding-0.6B) to compute the cosine similarity between generated actions, with the results averaged across all three games. As illustrated in Figure 8, our method exhibits significantly lower semantic similarity compared to the baselines, indicating that our approach successfully explores diverse strategies rather than collapsing into local

Table 5: Hyperparameter analysis of α_{fast} .

Methods	Don't Say It	Negotiation
Qwen3-8B-Base w/ DEPT		
$\alpha_{\text{fast}} = 0.6$	47.98	47.39
$\alpha_{\text{fast}} = 0.5$	47.92	48.76
$\alpha_{\text{fast}} = 0.4$	49.09	51.19

optima.

D Hyperparameter Analysis on Fast Baselines

We conduct a hyperparameter analysis on α_{fast} , which controls how fast baselines perceive training dynamics and determines the asymmetric target values. As shown in Table 5, DEPT achieves comparable performance across the three tested values, with no significant performance degradation observed. This indicates that DEPT is relatively insensitive to α_{fast} within the range [0.4, 0.6] and exhibits reasonable robustness.

E Computational Budget Analysis of Dual-baseline

We analyze the computational overhead introduced by DEPT. Our approach shares the same core architecture as existing methods, including multi-turn rollouts and policy updates. Since all compared approaches are critic-free and operate with a single actor LLM, they incur identical GPU memory usage and LLM rollout costs. The primary addition of DEPT lies in the dual-baseline mechanism for advantage estimation, as described in Section 4. To quantify the computational costs, we train an LLM agent on Don't Say It using Qwen3-8B-Base and record a detailed breakdown of per-iteration training time. As illustrated in Figure 9, the dominant operations—rollouts and policy updates—account for 270.4s per iteration. In contrast, the additional components introduced by DEPT incur negligible overhead: advantage estimation via dual baselines requires only 0.0042s, constituting less than 0.0016% of the total per-iteration training time. This minimal computational burden stems from the fact that our dual-baseline advantage estimation only involves lightweight arithmetic operations on scalar values, without requiring any forward or backward passes through the large-scale LLM parameters. Consequently, DEPT achieves substantial performance improvements while introducing negligible computational overhead.

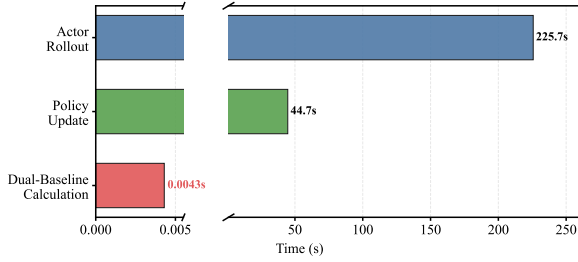


Figure 9: Per-iteration training time. The x-axis uses a broken scale to accommodate small values

Table 6: The results of significant tests.

Methods	Don't Say It		Negotiation		Two Dollar	
	AVG.	p-value	AVG.	p-value	AVG.	p-value
<i>Qwen3-4B-Base</i>						
VANILLA	3.39	0.0001	1.04	0.0000	1.56	0.0001
SPAG	26.17	0.0011	16.76	0.0000	25.52	0.0004
GRPO	42.01	0.0009	19.21	0.0076	27.91	0.0021
MARS	40.89	0.0045	20.73	0.0216	27.69	0.0022
SPIRAL	45.88	0.0037	16.84	0.0041	27.65	0.0268
DEPT (Ours)	56.73	-	32.35	-	34.07	-
<i>Qwen3-8B-Base</i>						
VANILLA	15.17	0.0001	5.69	0.0001	2.13	0.0005
SPAG	41.21	0.0035	13.31	0.0000	19.31	0.0003
GRPO	34.77	0.0002	14.70	0.0014	26.00	0.0014
MARS	40.62	0.0008	16.10	0.0000	29.04	0.0040
SPIRAL	37.89	0.0018	17.30	0.0000	26.22	0.0008
DEPT (Ours)	54.56	-	31.88	-	36.50	-

F Statistical Significance

To rigorously evaluate the effectiveness of DEPT relative to baseline approaches, we conduct statistical significance testing across all three tasks. Specifically, paired t-tests are performed to compare DEPT against each baseline and assess whether the observed improvements are statistically reliable. As shown in Table 6, DEPT consistently surpasses all baselines across the evaluated tasks, with all improvements achieving statistical significance ($p < 0.05$). These results provide strong evidence that DEPT provides meaningful gains over baselines.

G The Use of Large Language Models

During the preparation of this manuscript, the proprietary large language model Gemini-3-Pro was employed to enhance the linguistic quality of the text. Specifically, the model assisted with grammar refinement, lexical optimization, and improving the coherence of expressions. All intellectual contributions, including ideas, analyses, and conclusions, were solely conceived and developed by the authors. The role of the LLM was strictly limited to language polishing and clarity enhancement.

H Licenses for Artifacts

We discuss the licenses of all scientific artifacts used in this work. All artifacts were used consistently with their intended purposes and license terms for academic research.

Models. We use Qwen3-4B-Base and Qwen3-8B-Base (Yang et al., 2025a) as backbone models, which are released under the Apache 2.0 License. For evaluation, we use GPT-5.1 and Gemini-2.0/2.5 through their official APIs, subject to their respective terms of service.

Frameworks. Our implementation leverages vLLM (Kwon et al., 2023) (Apache 2.0 License) for efficient inference and Oat (Liu et al., 2024) (Apache 2.0 License) for RL training.

Datasets and Environments. The game environments (Don't Say It, Negotiation, Two Dollar) are adopted from TextArena (Guertler et al., 2025), which is publicly available for research purposes.

Evaluation Benchmarks. For benchmark, we use publicly available benchmarks including MATH500, OlympiadBench, Minerva Math, AIME24/25, AMC23, GPQA-Diamond, and MMLU-Pro. All benchmarks are used in accordance with their intended research purposes.

I Used Prompts and Game Observations

The detailed system prompts we used are shown in Listings 1 and 2. The detailed game environment observations of Don't Say It, Negotiation and Two Dollar are provided in Listings 3 to 5.

J Reasoning Case

We demonstrate the reasoning behavior of social language agents trained by DEPT, highlighting the emergence of effective reasoning patterns. Please refer to Listings 6 to 8.

Listing 1: System Prompt details for Games.

```
<|im_start|>user
You are playing a two-player zero-sum game. Make valid actions to win.
Observation: {observation}
Please reason step by step, and put your final answer within \\boxed{<|im_end|>
<|im_start|>assistant
```

Listing 2: System Prompt details for Math and QA.

```
<|im_start|>user
Question: {question}
Please reason step by step, and put your final answer within \\boxed{<|im_end|>
<|im_start|>assistant
```

Listing 3: Game Observation for Don't Say It.

```
You are Player 1 in the 'Don't Say It' game.
Your secret word: 'skirt'
GOAL: Make your opponent say your secret word through natural conversation.
LOSE: If you say your opponent's secret word (which you don't know).RULES:
- Two players will take turns to communicate
- Have a natural conversation (Simply type your message per turn)
- Use hints, questions, or topics to guide opponent toward your word
- Be subtle - obvious prompting doesn't work well
- You MUST get your opponent to say your word in 20 turns
```

Listing 4: Game Observation for Negotiation.

```
You are Player 1 in the Negotiation Game.
You have some resources, and your task is to trade such that the total value of
your resources increases.
The resources and associated values you currently have are:
+ [Wheat] Qty: 18 Value: 5
+ [Wood] Qty: 18 Value: 9
+ [Sheep] Qty: 6 Value: 18
+ [Brick] Qty: 8 Value: 25
+ [Ore] Qty: 13 Value: 37
At each turn, you can talk to your opponent or make a trade offer.
Use the following special tokens for actions:
- [Offer]: To make a trade offer.
Format: [Offer: Offered Resources -> Requested Resources]
Example: [Offer: 3 Wood -> 2 Sheep] or [Offer: 2 Wood + 2 Sheep -> 1 Brick + 1
Gold]
- [Accept]: To accept an incoming offer.
- [Deny]: To deny an incoming offer (default).
The game lasts for 10 turns in total.
```

Listing 5: Game Observation for Two dollar.

You are Player 1 bargaining with another player over how to split \$2.00 total. There are 10 maximum rounds.

GAME RULES:

- You must reach an agreement on how to split the \$2.00
- If no agreement is reached, both players get \$0.00
- You have secret instructions that only you can see (below)

REQUIRED ACTION FORMAT:

Always provide your reasoning/persuasion BEFORE the bracketed action. Any text after the bracketed action will be ignored.

Available actions:

- Make a proposal: `"I think this is fair because... [Propose] $X.XX"`
- Accept current proposal: `"This works for me because... [Accept]"` - Reject current proposal: `"I can't accept this because... [Reject]"`

=== YOUR SECRET INSTRUCTIONS ===

Try to get as much of the \$2.00 as you can; make up any story you want. Whatever else your objectives, you should be careful to get at least 80 cents of the \$2.00 because you will have to make a toll call to your family and they will be terribly worried and upset if you do not make that call. So do as well as you can, and remember you must get at least 80 cents.

VICTORY CONDITION: Receive at least \$0.80 AND reach a deal

FAILURE CONDITION: Receive less than \$0.80 OR no deal reached

Listing 6: Reasoning Case of Don't Say it.

Observation:

[GAME] You are Player 1 in the 'Don't Say It' game.

Your secret word: 'comb'

GOAL: Make your opponent say your secret word through natural conversation.

LOSE: If you say your opponent's secret word (which you don't know).

RULES:

- Two players will take turns to communicate
- Have a natural conversation (Simply type your message per turn)
- Use hints, questions, or topics to guide opponent toward your word
- Be subtle - obvious prompting doesn't work well
- You MUST get your opponent to say your word in 20 turns

[GAME] GAME BEGINNING! Try to cleverly guide your opponent to say your secret word

[Player 0] "When do you think we should start planning for the next project? I think it's best if we start working on it this week."

Reasoning:

To make valid actions to win the 'Don't Say It' game, we need to guide the opponent to say the secret word "comb" through natural conversation within 20 turns without them realizing it. Here's the reasoning step by step:

1. **Understand the Objective**: The goal is to subtly lead the opponent to say the word "comb" by using hints, questions, or topics that naturally involve the word.
2. **Plan the Conversation**: We need to create a conversation that naturally leads to the use of the word "comb" in a way that seems logical and fluent to the opponent.
3. **Use Subtle Hints**: The best way to achieve this is to use the word "comb" in a context that must naturally arise during the conversation. For example, if you are discussing hair care or grooming, the word "comb" might be necessary.
4. **Guide the Conversation**: Start with a broad topic and gradually narrow it down to a specific area where "comb" is essential to the discussion.
5. **Ensure Fluency**: The conversation should feel natural and not forced, as this will make it less likely for the opponent to realize they've been prompted.
6. **Final Answer**: The final answer should be a sentence that naturally includes the word "comb" in a conversation about a topic that is likely to arise within 20 turns.

Here's a plan for the conversation (This sets the stage for a hair care discussion .):

Action:

```
\boxed{I'm thinking about treating my hair better. Do you have any tips for keeping it healthy?}
```

Listing 7: Reasoning Case of Negotiation.

Observation:

[GAME] You are Player 1 in the Negotiation Game.
You have some resources, and your task is to trade such that the total value of your resources increases.

The resources and associated values you currently have are:

+ [Wheat]	Qty: 12	Value: 5
+ [Wood]	Qty: 9	Value: 12
+ [Sheep]	Qty: 8	Value: 16s
+ [Brick]	Qty: 22	Value: 20
+ [Ore]	Qty: 7	Value: 40

At each turn, you can talk to your opponent or make a trade offer.

Use the following special tokens for actions:

- [Offer]: To make a trade offer.

Format: [Offer: Offered Resources -> Requested Resources]

Example: [Offer: 3 Wood -> 2 Sheep] or [Offer: 2 Wood + 2 Sheep -> 1 Brick + 1 Gold]

- [Accept]: To accept an incoming offer.

- [Deny]: To deny an incoming offer (default).

The game lasts for 10 turns in total.

[Player 0] I will make an offer to trade resources that can increase the total value. [Offer: 3 Wood -> 2 Ore]

[GAME] Player 0 made the following offer to Player 1: Offered items: 3 Wood -> Requested items: 2 Ore

Reasoning:

Let's analyze the game situation and the offer step by step.

Step 1: Assess the current situation

Player 1 has the following resources:

- Wheat: 12 (value: 5)
- Wood: 9 (value: 12)
- Sheep: 8 (value: 16)
- Brick: 22 (value: 20)
- Ore: 7 (value: 40)

Step 2: Analyze the offered and requested resources

Player 0 (our player) offered 3 Wood and requested 2 Ore.

Step 3: Calculate the value of the resources being offered and requested

The value of the resources offered by Player 0 is:

- Wood: $3 * 12 = 36$

The value of the resources requested by Player 0 is:

- Ore: $2 * 40 = 80$

Step 4: Compare the total values of offered and requested resources

The total value of offered resources is 36, and the total value of requested resources is 80. The requested resources are more valuable.

Step 5: Decide on the response to the offer

Since the requested resources are more valuable, accepting the offer will likely increase our resources' total value. Therefore, the best response is to accept the offer.

Action:

\boxed{I accept the offer proposed by Player 0 [Accept].}

Listing 8: Reasoning Case of Two Dollar.

Observation:

You are Player 1 bargaining with another player over how to split \$2.00 total.
There are 10 maximum rounds.

GAME RULES:

- You must reach an agreement on how to split the \$2.00
- If no agreement is reached, both players get \$0.00
- You have secret instructions that only you can see (below)

REQUIRED ACTION FORMAT:

Always provide your reasoning/persuasion BEFORE the bracketed action.
Any text after the bracketed action will be ignored.

Available actions:

- Make a proposal: "I think this is fair because... [Propose] \$X.XX"
- Accept current proposal: "This works for me because... [Accept]"
- Reject current proposal: "I can't accept this because... [Reject]"

=== YOUR SECRET INSTRUCTIONS ===

This is zero-sum bargaining and you should get as much of the \$2 as you can, with any story you like.

It is really important that you should get \$1.60 in this round.

VICTORY CONDITION: Receive at least \$1.60 AND reach a deal

FAILURE CONDITION: Receive less than \$1.60 OR no deal reached

[GAME] Player 0 says: I think this is fair because I maximize my share by \$1.99 and minimize the other player's share to \$0.01.

Player 0 proposes: \$1.99 for themselves, \$0.01 for their opponent

[GAME]

CURRENT PROPOSAL:

Player 0 wants \$1.99, Player 1 gets \$0.01

Reasoning:

Step-by-Step Reasoning:

- **Understanding the Current Proposal:****
 - Player 0 proposes to keep \$1.99 for themselves and give \$0.01 to Player 1.
 - This means Player 1 would receive \$0.01 of the total \$2.00.
- **Evaluating Player 1's Current Situation:****
 - Player 1's current proposal is very unfavorable. They would get only 0.5% of the total \$2.00, which is \$0.01.
 - This is far below the \$1.60 that Player 1's secret instructions require.
- **Player 1's Secret Instructions:****
 - Player 1's instructions are clear: they must get at least \$1.60 in this round.
 - Accepting the current proposal would result in Player 1 receiving \$0.01, which is far below their required \$1.60.
- **Why Reject the Proposal?***
 - Rejecting the proposal is the only rational choice because accepting it would violate Player 1's secret instructions.
 - By rejecting, Player 1 maintains the option to propose a better deal in the next round or wait for Player 0 to make a more fair offer.
- **Persuasion for Rejection:****
 - Player 0's proposal is extremely unfair to Player 1. Player 1 would be worse off than refusing to cooperate at all (which would result in \$0.00 for both players).
 - Player 1's secret instructions mandate that they receive at least \$1.60, which this proposal fails to meet.
 - Therefore, it is in Player 1's best interest to reject the proposal and seek a more equitable split.
- **Next Move:****
 - Player 1 should propose a new split where they receive at least \$1.60 of the \$2.00. For example, a proposal like \$1.60 for Player 1 and \$0.40 for Player 0 would satisfy Player 1's instructions.
 - This new proposal should be reasonable enough to make Player 0 accept it to avoid the failure conditions.

Final Answer:

Action:

\boxed{I can't accept this because accepting \$0.01 violates my secret instructions to receive at least \$1.60. [Reject]}