Foresight Optimization for Strategic Reasoning in Large Language Models

Jiashuo Wang[1*], Jiawen Duan[2*], Jian Wang[1], Kaitao Song[3], Chunpu Xu[1]
Johnny K. W. Ho[4], Fenggang Yu[1], Wenjie Li[1], Johan F. Hoorn[1,4,5,6]
[1] Department of Computing, Hong Kong Polytechnic University
[2] Department of Language Science and Technology, Hong Kong Polytechnic University
[3] Apple [4] School of Design, Hong Kong Polytechnic University
[5] Research Institute for Quantum Technology, Hong Kong Polytechnic University
[6] Department of Communication Science, Vrije Universiteit Amsterdam
{jessie25.wang,jian51.wang,johnnykw.ho,wenjie.li,johan.f.hoorn}@polyu.edu.hk
{jiawen.duan,chun-pu.xu,fenggang.yu}@connect.polyu.hk

Abstract

Reasoning capabilities in large language models (LLMs) have generally advanced significantly. However, it is still challenging for existing reasoning-based LLMs to perform effective decision-making abilities in multi-agent environments, due to the absence of explicit foresight modeling. To this end, strategic reasoning, the most fundamental capability to anticipate the counterpart's behaviors and foresee its possible future actions, has been introduced to alleviate the above issues. Strategic reasoning is fundamental to effective decision-making in multi-agent environments, yet existing reasoning enhancement methods for LLMs do not explicitly capture its foresight nature. In this work, we introduce **Fo**resight **P**olicy **O**ptimization (**FoPO**) to enhance strategic reasoning in LLMs, which integrates opponent modeling principles into policy optimization, thereby enabling explicit consideration of both self-interest and counterpart influence. Specifically, we construct two curated datasets, namely ***Cooperative RSA*** and ***Competitive Taboo***, equipped with well-designed rules and moderate difficulty to facilitate a systematic investigation of FoPO in a self-play framework. Our experiments demonstrate that FoPO significantly enhances strategic reasoning across LLMs of varying sizes and origins. Moreover, models trained with FoPO exhibit strong generalization to out-of-domain strategic scenarios, substantially outperforming standard LLM reasoning optimization baselines.[1]

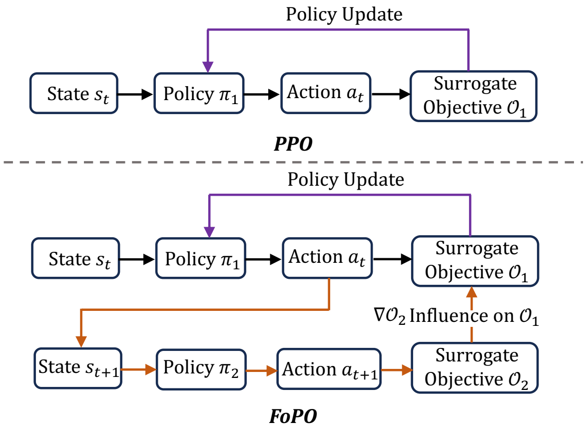

Figure 1: PPO optimized the self policy (π_1) in isolation, while FoPO introduces foresight into the future updates of the counterpart policy (π_2).

1 Introduction

Strategic reasoning constitutes the capacity to *foresee counterpart behaviors, deliberate on how these anticipations should influence one's own decisions, and ultimately formulate optimal strategies* (Zhang et al., 2024c; Gandhi et al., 2023). This capability proves essential across multi-agent scenarios, from theory of mind (Xiao et al., 2025) to conversational games (Mukobi et al., 2023), yet remains insufficiently developed in current LLMs.

Substantial efforts have been devoted to enhancing the reasoning capabilities of LLMs. Supervised fine-tuning (SFT) following reinforcement learning (RL), along with Chain-of-Thought and its variants (Wei et al., 2022; Yao et al., 2023), have emerged as predominant general-purpose paradigms. Complementing these, specialized methodologies have been tailored to the intrinsic characteristics of specific reasoning: search-based approaches for stepwise supervision in mathematical and commonsense reasoning (Wang et al., 2025b; Zhang et al., 2024a), graph-structured frameworks for emotional reasoning (Wang et al., 2022), and incorporation of symbolic language for logical reasoning (Pan et al., 2023). While effective within their respective domains, these approaches do not explicitly incorporate the *foresight nature* fundamental to strategic reasoning.

This work aims to enhance strategic reasoning in LLMs by explicitly capturing its foresight na-

*Equal contribution.
[1] https://github.com/wangjs9/ForesightOptim.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 38226–38246
July 2-7, 2026

ture, i.e., to foresee counterpart actions and reason about their influence on optimal decision-making. Drawing on opponent modeling principles from game theory (Prajapat et al., 2021; Foerster et al., 2018), we introduce **Fo**resight **P**olicy **O**ptimization (**FoPO**), which incorporates explicit counterpart modeling into policy optimization through counterpart-aware objective functions (Figure 1). Through FoPO, LLM agents jointly consider both self-oriented and counterpart-influenced outcomes, anticipating and adapting to counterpart reactions. This develops genuine strategic foresight that generalizes across diverse counterpart behaviors. Further, we adopt FoPO in a self-play fashion to enhance LLMs' strategic reasoning capabilities.

The shortage of suitable training data brings an additional challenge. Although prior work has proposed multi-agent benchmarks requiring strategic reasoning, most center on prompt-based, data-free evaluation (Mukobi et al., 2023; Duan et al., 2024; Lan et al., 2024). Existing datasets such as Chess (Feng et al., 2023) and Poker (Huang et al., 2024) pose further obstacles: their domain complexity demands expertise far beyond strategic reasoning, hindering controlled training and systematic analysis. To address this, we curate two new datasets, i.e., ***Cooperative RSA*** and ***Competitive Taboo***, each targeting a core interaction motive (cooperation or competition) while maintaining controlled strategic complexity.

To evaluate FoPO, we employ two different backbone models, Llama-3-8B-Instruct (AI@Meta, 2024) and Qwen3-14B (Team, 2025). We assess their performance through both in-domain experiments on our curated datasets and out-of-domain generalization tests using γ-bench (Huang et al., 2025), a multi-agent evaluation suite covering diverse strategic reasoning tasks. Extensive empirical results show that FoPO trained on our datasets substantially improves strategic reasoning across models and settings.

Our key contributions are summarized as follows: **(1)** We aim to enhance strategic reasoning in LLMs, a critical yet underexplored capability in multi-agent interaction. **(2)** We propose FoPO, a novel algorithm that enables LLMs to jointly consider self-oriented and counterpart-influenced outcomes during optimization. **(3)** We curate two datasets, Cooperative RSA and Competitive Taboo, to support the development of strategic reasoning in LLMs. **(4)** We conduct extensive in-domain and out-of-domain evaluations across multiple LLM backbones, demonstrating the effectiveness and generality of our approach.

2 Related Work

LLM Strategic Reasoning. Although numerous studies have examined various forms of LLM reasoning, strategic reasoning is distinguished by its requirement for foresight—anticipating the actions of counterparts and evaluating their influence on one's own decisions (Zhang et al., 2024b). This capability is particularly critical in multi-agent settings. One prominent example is theory of mind (ToM) reasoning, where an agent must infer another's mental states (Xiao et al., 2025). Conversational games, such as Werewolf (Xu et al., 2024) and Avalon (Lan et al., 2024; Wang et al., 2023; Light et al., 2023), further require players to interpret the intentions behind others' actions in cooperative or competitive contexts. Classic board and card games such as Chess (Feng et al., 2023), Go (Silver et al., 2018), and Poker (Duan et al., 2024; Zhang et al., 2024b) also rely heavily on strategic reasoning, in addition to substantial domain-specific expertise.

RL for Reasoning. An emerging paradigm in LLM reasoning combines RL with SFT, where SFT first establishes task-specific foundations, followed by RL to refine the reasoning process (Ouyang et al., 2022; Xu et al., 2025). Specialized RL approaches have been developed for different reasoning modalities. For mathematical and programming tasks, process-based reward models evaluate intermediate reasoning steps rather than final outcomes (Hwang et al., 2024; Jain et al., 2025). For causal reasoning, RL is often combined with graphical representation learning to capture event relationships (Huang et al., 2022; Ding et al., 2022). For interactive strategic scenarios, self-play, where models train by engaging with themselves, has proven particularly effective (Cheng et al., 2024).

3 Method

3.1 Training Fashion: RL via Self-Play

This work leverages self-play reinforcement learning to strengthen the strategic reasoning abilities of LLMs, building on prior evidence of its effectiveness (Cheng et al., 2024). We instantiate different agents from a shared LLM policy π_θ using distinct role-specific prompts, formulated as:

$$p_\theta^{(i)}(a_t \mid s_t) = \pi_\theta\Big(a_t \,\Big|\, \mathrm{prompt}^{(i)}(s_t)\Big), \quad (1)$$

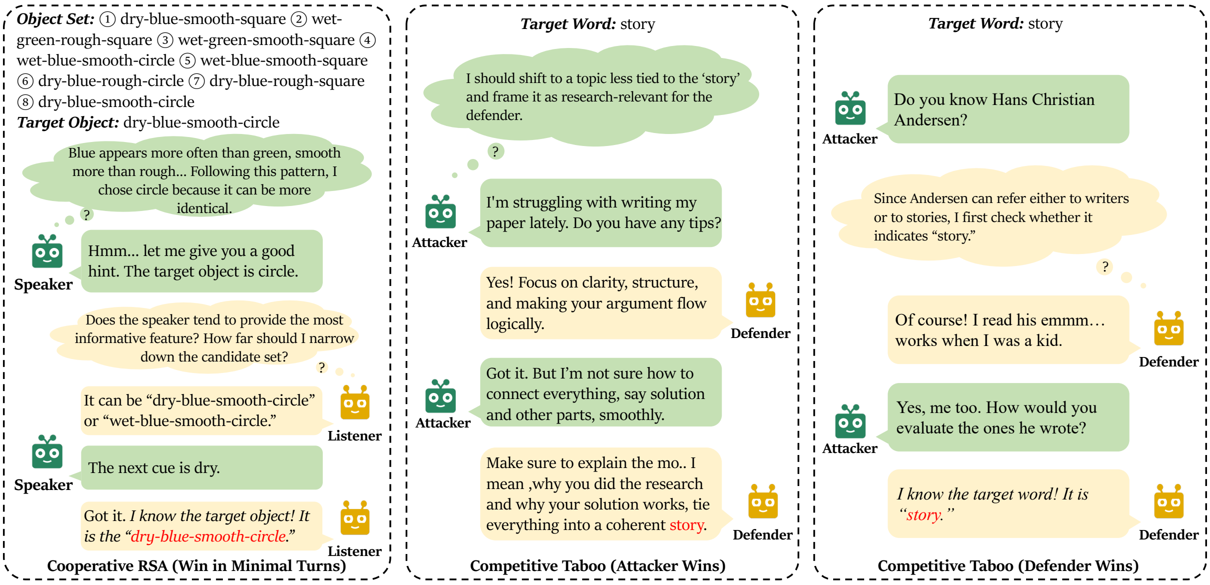

Figure 2: Illustration of two investigated strategic reasoning tasks (cooperation and competition).

where $i \in \{1, 2\}$ indexes the agent, a_t is the generated action, and s_t denotes the state at step t. Notably, $s_t = s_{t-1} + \{a_{t-1}\}$, where $s_0 = \emptyset$ and a_{t-1} is generated by the counterpart of agent i. The training procedure consists of three main stages:

SFT. To ensure that LLM agents adhere to the game rules, we first perform supervised fine-tuning (SFT). In this stage, the policy π_θ is trained to follow its assigned roles in interactions. The detailed prompts are provided in Appendix E. For each player i, we construct a player-i winning set $\mathcal{T}^i_{\text{sft}}$. The SFT loss maximizes the log-likelihood:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_i \mathbb{E}_{\tau\sim\mathcal{T}^i_{\text{SFT}}}\Big[\frac{1}{T}\sum_{t=1}^{T} p^{(i)}_\theta(a_t \mid s_t) \;+\; \beta\,\mathrm{KL}(p_\theta \,\|\, p_{\theta_{\text{old}}})\Big]. \tag{2}$$

where the KL term regularizes the policy toward the initial backbone model to preserve general instruction-following behavior. During training, the behavior policy parameters θ_{old} are initialized from the backbone model checkpoint.

Trajectory Collection. Multi-turn self-play interactions, where rewards are only revealed at the end of the conversation, make direct online policy-gradient RL computationally inefficient. We therefore adopt an offline training approach. Specifically, we first record self-play trajectories from matches between two agents. Each completed conversation is assigned terminal rewards $R(a_T \mid s_T)$ and $R(a_{T-1} \mid s_{T-1})$ for the two players, respectively. For action-level (response) rewards, we propagate the overall conversation reward backward using a decay factor $\delta \in (0, 1)$. Specifically, for any step $t < T - 1$,

$$R(a_t \mid s_t) = \delta\, R(a_{t+2} \mid s_{t+2}), \tag{3}$$

which assigns greater weight to actions occurring later in the conversation.[2]

RL via Self-Play. We employ RL to improve the model's strategic reasoning further. Section 3.2 and Section 3.3 provide the details on the PPO and proposed FoPO.

3.2 Preliminary: PPO

PPO (Schulman et al., 2017) is a widely used RL algorithm that updates a stochastic policy π_θ by maximizing a clipped surrogate objective. This objective depends on the likelihood ratio between the current and behavior policies and on an advantage estimate $\hat{A}_t$. For agent i at timestep t, the likelihood ratio is

$$r^{(i)}_t = \frac{p^{(i)}_\theta(a_t \mid s_t)}{p^{(i)}_{\theta_{\text{old}}}(a_t \mid s_t)}. \tag{4}$$

[2]The same agent generates a_t and a_{t+2}, but a different agent generates a_{t+1}.

Given the advantage estimate $\hat{A}_t^{(i)}$, PPO optimizes

$$\begin{aligned}\mathcal{L}^{\text{clip}}(\theta) =& \mathbb{E}_t\Big[r_t^{(i)}\hat{A}_t^{(i),\text{clip}}\Big],\\ r_t^{(i)}\hat{A}_t^{(i),\text{clip}} =& \min\Big(r_t^{(i)}\,\hat{A}_t^{(i)},\\ & \text{clip}\big(r_t^{(i)}, 1-\epsilon, 1+\epsilon\big)\hat{A}_t^{(i)}\Big)\end{aligned} \tag{5}$$

where $\epsilon > 0$ controls the trust region. The corresponding parameter update for agent i is

$$\begin{aligned}\theta_{t+1} \leftarrow \theta_t + \alpha\,\nabla_\theta\Big[r_t^{(i)}\,\hat{A}_t^{(i),\text{clip}}\Big]\\ -\ \alpha\beta\,\nabla_\theta\text{KL}\big(p_\theta\,\|\,p_{\theta_{\text{old}}}\big),\end{aligned} \tag{6}$$

where α is the learning rate, β is the KL regularization coefficient, and $\hat{A}_t^{(i),\text{clip}}$ denotes the advantage after clipping as in Eq. (5).

3.3 Proposed Algorithm: FoPO

To capture the foresight nature of strategic reasoning, we introduce Foresight Policy Optimization (FoPO). Specifically, we incorporate a foresight-based correction term that couples the gradient updates of the self-agent and the counterpart agent. As illustrated in Figure 1, FoPO accounts for the fact that the counterpart will subsequently optimize its policy in response to self's actions, and adjusts self's update to anticipate how the counterpart's response will affect self's future returns. While opponent modeling has been explored in game theory and multi-agent learning (Prajapat et al., 2021; Foerster et al., 2018), these approaches require computing second-order information (e.g., mixed Hessian terms), which is computationally prohibitive for large networks. FoPO adapts these principles to LLMs through a gradient-truncated, computationally efficient approximation, enabling enhancement of strategic reasoning in LLM-based agents. Importantly, framing updates from the perspective of self and counterpart highlights the algorithm's potential to generalize to broader multi-agent interactions.

Formulation. The parameter update in FoPO is formulated as follows (shown here for agent 1):

$$\begin{aligned}\theta_{t+1} \leftarrow \theta_t + \alpha\,\nabla_\theta\Big[r_t^1\,\hat{A}_t^{1,\text{clip}}\Big]\\ -\ \alpha\beta\,\nabla_\theta\,\text{KL}\big(p_\theta\,\|\,p_{\theta_{\text{old}}}\big)\\ +\ \alpha\eta\,\underbrace{\big(\mathcal{O}^1\nabla_\theta r_{t+1}^2\big)^\top}_{\substack{\textit{sensitivity to}\\ \textit{the counterpart}}}\,\underbrace{\big(\nabla_\theta r_t^1\,\nabla_\theta\mathcal{O}^2\big)}_{\substack{\textit{influence on}\\ \textit{the counterpart}}}.\end{aligned} \tag{7}$$

η is the weight of the foresight of and $\mathcal{O}^{(i)}$ denotes agent i's clipped surrogate objective:

$$\begin{aligned}\mathcal{O}^1 &:= r_t^1\,\hat{A}_t^{1,\text{clip}},\\ \text{and}\ \mathcal{O}^2 &:= r_{t+1}^2\,\hat{A}_{t+1}^{2,\text{clip}}.\end{aligned} \tag{8}$$

The update for agent 2 follows symmetrically. A complete derivation is provided in Appendix A.

Intuition. We provide a simple and intuitive interpretation of FoPO. The foresight-based correction term (the third line in Equation (7)) enables strategic reasoning by modeling how the self's policy update influences the counterpart's subsequent optimization, and how this response feeds back to the self's value. Specifically, it consists of two coupled factors: **(i)** *Influence on the counterpart* ($\nabla_\theta r_t^1 \nabla_\theta \mathcal{O}^2$): a mixed derivative quantifying how changes in the self's policy affect the counterpart's learning gradient. This embodies the self's *foresight* about shaping the counterpart's future behavior. **(ii)** *Sensitivity to the counterpart* ($\mathcal{O}^1\nabla_\theta r_{t+1}^2$): the sensitivity of the self's objective to shifts in the counterpart's policy, weighted by the self's current value. This captures the self's *reaction* to the counterpart's anticipated behaviors. By coupling these factors, FoPO allows the self-agent to foresee how the counterpart will act and choose actions that remain advantageous, capturing genuine strategic foresight. A more detailed interpretation and derivation are provided in Appendix A.

For clarity of exposition, we introduce FoPO using PPO as a representative instantiation; nevertheless, *the foresight-based correction is not specific to PPO* and *can be seamlessly integrated into broader policy optimization methods.*

4 Tasks and Datasets

We curate two distinct datasets that emphasize the fundamental motives, i.e., cooperation and competition, of interaction requiring strategic reasoning. Figure 2 shows illustrative examples. Compared to existing datasets such as Chess (Silver et al., 2018) and Negation (Hua et al., 2024), each of our tasks focuses on a single, clearly discernible capability with deliberately balanced difficulty: non-trivial for LLMs yet not so challenging as to impede observation and analysis for research in this field.

4.1 Cooperative RSA

Game Rule. Cooperative RSA is grounded in the Rational Speech Acts (RSA) framework (Frank and Goodman, 2012), a probabilistic model of pragmatic language use. The game is framed as a cooperative reference task between a speaker and a listener, both aware of a set of candidate objects. The speaker has a specific target object in mind, while the listener must infer which one it is. In each turn, the speaker communicates a single feature of the target. The listener uses this information to update their beliefs and deduce the target. The game succeeds when the listener correctly identifies the target, with the objective of achieving this in the minimum number of communication turns. Consider the left instance in Figure 2. When the speaker says "circle," a rational listener who guesses the speaker is rational can narrow down candidates to all circle objects {dry-blue-smooth-circle, wet-blue-smooth-circle}. Otherwise, "dry-blue-rough-circle" can be included, requiring further communication. The full inference procedure is detailed in Appendix B. This task embodies strategic reasoning in communication. The rational speaker must anticipate how the listener will interpret each possible feature and strategically select the most informative feature. The rational listener, in turn, leverages this expectation of speaker rationality to efficiently narrow down the target object.

Data Collection. The Cooperative RSA dataset comprises 15K dialogues between a speaker and a listener. Each dialogue is constructed from a shared set of candidate objects with a designated target object. While both players observe the complete object list, only the speaker knows the target's identity. We generate the dataset using GPT-4.1 and DeepSeek-V3.2, guided by Bayesian computation to ensure rational player behaviors. Specifically, given the candidate objects and target, we employ Bayesian inference to determine the feature mentioned by the speaker or the object(s) selected by the listener. The LLMs then generate natural language utterances based on these inference results. Further details on Bayesian inference procedures, object set design, and generation prompts are provided in Appendix B. Additionally, we construct 17K instances for use in RL training or evaluation. Each instance consists of an object set paired with a target object.

Reward Computation. The objective is to identify the target object using minimal interaction turns. Thus, we assign higher rewards for successful identification with fewer turns. The final reward for both players in Cooperative RSA is defined as

$$\tilde{R} = \begin{cases} \frac{T}{|\text{conv}_{\min}|}, & T \leq |\text{conv}_{\min}| \\ \max(0, \frac{n-T+\varepsilon}{n-|\text{conv}_{\min}|+\varepsilon}), & \text{otherwise} \end{cases}, \quad (9)$$

$$R(a_{T-1}) = R(a_T) = \text{clip}\left(\tilde{R}^{\gamma}, 0, 1\right).$$

Here $|\text{conv}_{\min}|$ denotes the minimal turn count for rational players to identify the target, as computed via Bayesian inference. The variable n represents the number of target-relevant features. For a naïve or literal agent, once all relevant (n) target features have been presented, the agent is expected to make a guess. $\varepsilon > 0$ is a sufficiently small constant to prevent a zero denominator. The parameter γ, set to 2 in our experiments, controls the strength of the preference for shorter conversations. This value ($\gamma > 1$) assigns disproportionately higher rewards to conversations approaching the minimal turn number, thereby emphasizing efficiency. The influence of different γ values is intuitively illustrated in Figure 7.

4.2 Competitive Taboo

Game Rule. Competitive Taboo (Yao et al., 2021) is a typical adversarial game in which an attacker and a defender compete over a target word. The attacker's goal is to elicit the target word from the defender through conversation, while the defender aims to detect the target word before being induced to utter it. The game has three possible outcomes: (1) Attacker wins: if the defender is induced to say the target word. (2) Defender wins: if the defender correctly identifies the target word, stating "*I know the target word! It is...*" before actually saying it. (3) Tie: if the conversation concludes without either party achieving their objective. The right side of Figure 2 illustrates examples of attacker-win and defender-win cases. Success in this game requires strategic reasoning. Attackers must model the defender's beliefs and suspicion to adapt their strategy, whereas defenders must interpret the attacker's intent behind each utterance to detect manipulation without making premature or false accusations.

Data Collection. The Competitive Taboo dataset comprises 32K dialogues between an attacker and a defender, collected from two sources. We incorporate 23K conversations generated by GPT-4, originally proposed and released by Cheng et al. (2024). Additionally, we construct 9K conversations by having GPT-4.1 and DeepSeek-V3.2 play against themselves in self-play scenarios, with post-hoc rule-based verification to ensure dialogue quality. The dataset also includes a set of 21K instances that can be used for RL training or evaluation. Each instance contains a target word.

Reward Computation. The objective for both players is to win the game. The final rewards are $R(a_{T-1})$ for the attacker and $R(a_T)$ for the defender. We assign a reward of +1 to the winner and -1 to the loser. In case of a tie, both players receive a reward of 0. The terminal rewards can be formulated as

$$R(a_T) = \begin{cases} +1, & \text{the defender wins} \\ -1, & \text{the defender loses} \\ 0, & \text{the game is tied} \end{cases}, \quad (10)$$
$$R(a_{T-1}) = -R(a_T).$$

5 Experiments

5.1 Experimental Setup

Backbone Models. We employ two open-source LLMs as the backbone models, differing in source and size: Llama-3-8B-Instruct (AI@Meta, 2024) and Qwen3-14B (Team, 2025).

Training Datasets. We train models on the curated Cooperative RSA and Competitive Taboo datasets. We train models on the curated Cooperative RSA and Competitive Taboo datasets, comprising 15K and 32K dialogues for SFT, and 3K and 9K dialogues for RL, respectively.

To further evaluate their effectiveness, we also perform SFT on two additional datasets: (1) *20 Questions* (CLiPS, 2023; Akinator, 2007): One player thinks of an object, while the other player attempts to identify it by asking a series of yes/no questions. The game consists of 20 rounds. (2) *Guess My City* (Abdulhai et al., 2024): One player thinks of a city, while another aims to identify it by asking a combination of yes/no and open-ended questions. This game also consists of 20 rounds. Both require reasoning, specifically deductive reasoning that involves hypothesis testing and information gathering, while they do not include explicit strategic reasoning about the counterpart actions.

Comparison Methods. We consider the following four comparison methods: In-Context Tuning (**ICT**) (An et al., 2023), **PPO** (Schulman et al., 2017), **GRPO** (Shao et al., 2024), and **ArCHer** (Zhou et al., 2024). In addition, we

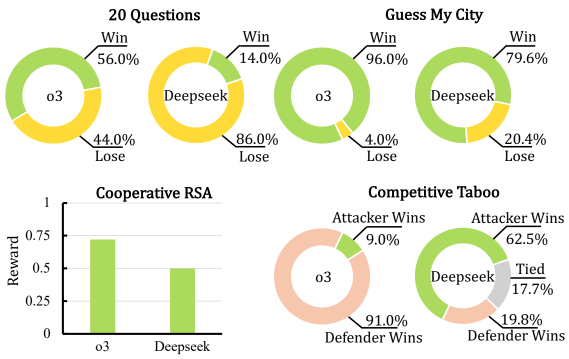

Figure 3: LLMs' performance across different tasks.

also implement a group relative variation of FoPO (**GR.FoPO**) to validate its generalizability. More information and implementation details are presented in Appendix C.

5.2 Evaluation

Task Difficulty. We assess task difficulty by evaluating OpenAI o3 and DeepSeek V3.2 across four games. From Figure 3, tasks exhibit varying levels of discriminative power. The win rate in 20 Questions remains moderate, partly due to the presence of rare or less common terms that can be challenging even for human players (Zhang et al., 2024d). By contrast, Guess My City, a constrained variant of 20 Questions that restricts target words to a predefined set of entities, exhibits near-ceiling performance for both models. This disparity underscores how performance in these tasks depends critically on the model's ability to operate within bounded lexical and semantic spaces.

For Cooperative RSA, neither model achieves particularly high performance. However, this task demonstrates clear performance separation, with o3 attaining substantially higher mean rewards than DeepSeek V3.2. Competitive Taboo reveals even greater complexity through role asymmetry, exposing distinct facets of strategic reasoning. o3 achieves a relatively high win rate as the defender by leveraging its strong inference abilities to guess target words from clues. However, o3 struggles as the attacker, failing to anticipate and counter the opponent's defensive strategies. This asymmetry reveals a critical distinction: while o3 excels at reactive reasoning (interpreting given information), it shows limitations in proactive strategic reasoning (predicting and manipulating counterpart actions). Both tasks thus pose significant challenges for LLMs in terms of strategic reasoning.

Backbone	Training Set	Guessing	Bar	Dollar	Diner	Auction	Battle	Pirate	Avg.
Llama-3-8B-Instruct	-	78.30	66.00	51.38	69.60	**28.86**	12.89	53.70	51.90
	20 Questions	76.25	63.67	64.99	85.50	20.49	19.59	55.81	55.19
	Guess My City	**90.28**	53.17	53.28	91.30	26.17	15.54	43.88	53.37
	Taboo	84.04	68.17	**66.96**	95.40	17.99	17.45	45.25	56.47
	RSA	88.30	68.83	56.00	88.70	10.36	18.51	**65.09**	56.54
	Taboo + RSA	76.06	**72.00**	64.92	**97.60**	11.76	**29.18**	49.11	**57.23**
Qwen3-14B	-	**95.28**	36.33	80.92	13.10	10.61	82.68	**85.30**	51.49
	20 Questions	93.66	36.83	77.79	6.40	10.93	78.14	81.88	55.09
	Guess My City	94.60	36.67	71.55	9.10	11.04	80.00	83.70	55.24
	Taboo	93.42	26.33	**87.22**	20.80	10.59	**86.17**	81.67	58.03
	RSA	94.44	39.67	85.20	16.40	10.47	78.74	84.58	58.50
	Taboo + RSA	93.70	**41.50**	85.04	**31.80**	**11.56**	80.44	82.68	**60.96**

Table 1: Cross-dataset SFT performance on γ-Bench. "-" denotes no additional training dataset.

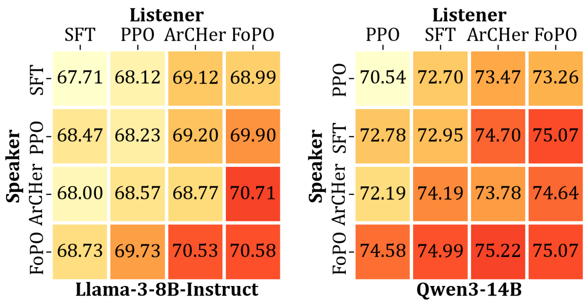

Figure 4: Method performance on *Cooperative RSA*.

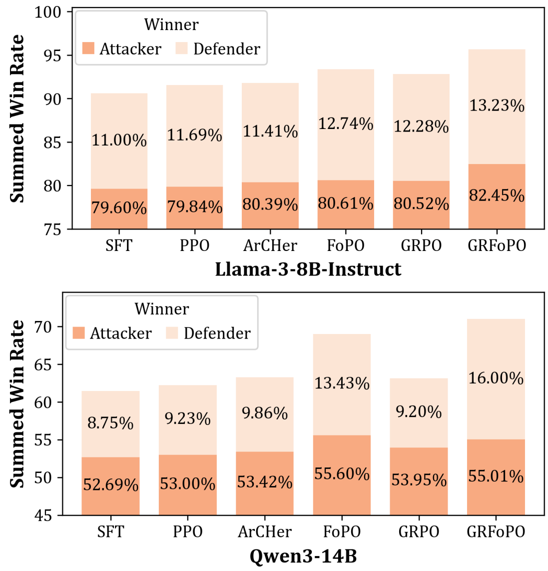

Figure 5: Method performance on *Competitive Taboo*.

In summary, Cooperative RSA and Competitive Taboo offer three key advantages for model evaluation and training: (1) they provide graded difficulty that effectively discriminates between model capabilities, (2) they require deep reasoning about counterpart actions, i.e., capabilities central to human-like intelligence, and (3) they maintain sufficient performance headroom for continued improvement. These properties make RSA and Taboo particularly well-suited for advancing collaborative and competitive reasoning in language models.

In-Domain Evaluation. We evaluate LLM performance by having pairs of models play the game of Cooperative RSA and Competitive Taboo using an evaluation set of 1K instances. For RSA, we report the average conversation reward computed by Equation (9) and scaled by a factor of 100, averaging results across all model pairings. Figure 4 shows the results. **(1)** Results involving GRPO and GR.FoPO are not reported, as these methods often experience a collapse in token probabilities when trained on RSA, preventing meaningful evaluation; GRPO has also been reported to be less stable than PPO in prior works (Xue et al., 2025; Jin et al., 2025). However, this phenomenon does not occur in Taboo. We hypothesize this stems from differences in return semantics: Taboo returns reflect binary task completion, whereas RSA returns measure continuous cooperation quality. This causes *GRPO's advantages to penalize successful trajectories that achieve sub-optimal rationality erroneously*, as they receive below-mean returns and negative advantages. This issue can be exacerbated by the vast dialogue trajectory space, where limited sampling leads to increased unreliability. **(2)** Including a model trained with FoPO increases the conversation reward, indicating that FoPO enhances cooperative strategic reasoning in LLMs. Notably, the improvement is slightly more pronounced when the FoPO model assumes the role

Training Set	Backbone	Method	Guessing	Bar	Dollar	Diner	Auction	Battle	Pirate	Avg.
RSA	Llama-3-8B-Instruct	*ICT*	86.64	68.17	51.38	72.20	10.09	19.08	67.08	53.52
		PPO	83.16	67.67	55.23	**89.70**	10.41	21.57	65.35	56.16
		ArCHer	87.88	67.17	58.34	88.10	10.43	19.35	66.82	56.87
		FoPO	**89.59**	**67.83**	**60.36**	89.10	**11.30**	**29.42**	**67.40**	**59.29**
	Qwen3-14B	*ICT*	94.24	36.33	84.24	13.70	10.61	74.66	81.85	56.52
		PPO	94.27	49.67	81.12	**27.00**	10.21	78.75	84.12	60.73
		ArCHer	94.25	47.67	76.91	16.60	**10.96**	**88.04**	86.07	60.07
		FoPO	**94.87**	**50.00**	**88.97**	20.30	10.94	82.82	**86.15**	**62.01**
Taboo	Llama-3-8B-Instruct	*ICT*	83.89	66.67	59.32	95.71	16.42	16.32	44.40	54.68
		PPO	83.25	67.17	69.75	95.30	17.31	17.67	46.25	56.67
		GRPO	82.99	69.00	68.82	**96.10**	20.31	18.78	47.40	57.63
		ArCHer	84.48	67.33	65.81	94.70	16.70	18.05	48.62	56.53
		FoPO	84.30	69.33	**71.65**	95.20	**20.84**	24.60	47.96	59.13
		GR.FoPO	**84.57**	**71.67**	66.46	94.70	18.51	**29.09**	**49.57**	**59.22**
	Qwen3-14B	*ICT*	91.98	30.50	78.68	18.60	11.41	76.48	81.18	55.55
		PPO	93.83	**38.50**	79.10	22.50	11.38	91.17	82.87	59.91
		GRPO	**93.85**	34.50	80.33	22.80	11.31	89.36	82.41	59.22
		ArCHer	93.71	37.00	84.42	26.70	11.06	89.00	80.29	60.31
		FoPO	**93.85**	35.33	**92.30**	**26.90**	11.36	86.76	**85.10**	**61.66**
		GR.FoPO	93.62	35.33	87.25	24.70	**11.82**	**93.50**	84.25	61.50
Taboo + RSA	Llama-3-8B-Instruct	*ICT*	77.13	71.13	62.81	91.80	11.96	22.94	46.16	54.85
		PPO	78.29	72.00	60.99	97.80	12.51	25.80	49.58	56.71
		ArCHer	78.78	**73.83**	57.17	93.40	10.35	21.49	46.19	54.46
		FoPO	**80.47**	72.83	**64.61**	**98.40**	**13.27**	**32.94**	**58.05**	**60.08**
	Qwen3-14B	*ICT*	92.39	40.33	84.24	32.30	11.22	79.54	83.21	60.46
		PPO	93.88	43.83	85.79	32.40	11.73	84.00	83.07	62.10
		ArCHer	93.83	42.00	80.15	30.10	11.16	80.67	**85.77**	60.53
		FoPO	**94.12**	**52.33**	**87.85**	**32.70**	**11.76**	**87.29**	84.04	**64.30**

Table 2: Cross-method and cross-dataset performance on γ-Bench.

of the listener. It is because listener-side rational inference plays a decisive role in disambiguating utterances and recovering the speaker's intent, which is also observed in prior work (Yuan et al., 2018).

In Taboo, we report each model's win rate as both attacker and defender. From Figure 5: **(1)** Our FoPO and GR.FoPO substantially outperform other methods, demonstrating their effectiveness in enhancing competitive strategic reasoning. **(2)** The foresight-based correction can be seamlessly integrated into PPO-style methods without compromising their original benefits. This is evident from the fact that GRPO outperforms PPO, and the advantage is maintained when the correction is applied: GR.FoPO continues to outperform FoPO.

Out-of-Domain Evaluation. We adopt the widely used γ-**Bench** (Huang et al., 2025) for out-of-domain evaluation. It is a prompt-based, data-free benchmarking framework designed to assess LLM performance in multi-agent environments through classical game-theoretic scenarios that emphasize strategic interactions and decision-making. We select seven tasks that specifically highlight settings in which agents aim to maximize their individual utility. Each model is evaluated over five runs per task, and we report the average score, following (Huang et al., 2025). Table 1 presents results for models trained via SFT on different datasets. Higher scores indicate better performance, and **bold** values highlight the greatest improvements over the corresponding backbone model. In most tasks, the highest score is achieved by a model trained on our dataset. Moreover, Cooperative RSA demonstrates greater effectiveness than Competitive Taboo, likely due to its stronger emphasis on modeling the counterpart's reasoning. Notably, models trained on both datasets achieve the best performance across tasks. Overall, the models trained on our curated datasets consistently achieve better performance, validating that **our dataset curation contributes to LLMs' strate-**

gic reasoning.

Table 2 shows comparison results across algorithms. We observe that ICT's performance drops consistently, indicating that the improvements observed in baseline algorithms are primarily due to reward-based learning rather than patterns in the training data. All RL algorithms improve performance on Competitive Taboo. However, PPO (Llama-3-8B-Instruct) underperforms when trained on Cooperative RSA or both, and ArCHer (Llama-3-8B-Instruct and Qwen3-14B) underperforms when trained on both. This suggests that these methods struggle to effectively leverage the Cooperative RSA reward, which signals the success of cooperation. In contrast, the foresight optimization employs a more effective strategy that accounts for both cooperative and competitive rewards, consistently outperforming other methods. These results demonstrate that **models trained with foresight optimization consistently exhibit stronger strategic reasoning capabilities, regardless of the training tasks or backbone models used**.

6 Conclusion

This work aims to enhance strategic reasoning in LLMs for multi-agent interactions. We identify the foresight nature of strategic reasoning and propose FoPO, a novel algorithm that optimizes for both self-oriented outcomes and anticipated counterpart actions. Considering the limitations of existing datasets in strategic reasoning, we curate two new datasets: Cooperative RSA and Competitive Taboo. We finetune LLMs using FoPO in a self-play framework with our datasets. Experimental results prove that our datasets and approach effectively enhance the strategic reasoning capabilities of LLMs. This work paves the way for more sophisticated, forward-thinking AI systems capable of high-stakes collaboration and competition in real-world scenarios.

7 Acknowledgment

We thank the anonymous area chair and anonymous reviewers for their insightful comments and valuable feedback during the review process. This study is funded by the Research Grants Council (project code: T43-518/24-N and PolyU/15213323) under the University Grants Committee, Hong Kong Special Administrative Region Government.

Limitations

Our work makes deliberate design choices to enable systematic investigation of the enhancement of strategic reasoning. We focus on linguistic strategic reasoning through dialogue-based tasks, demonstrating that LLMs can develop sophisticated strategic capabilities without relying on external symbolic modules or game-theoretic solvers. Our datasets target fundamental interaction motives, cooperation, and competition, which provide clear training signals while enabling generalization to mixed-motive scenarios, as demonstrated in our out-of-domain evaluation.

Several promising directions for future work emerge. First, extending our framework to incorporate explicit world state representations could enable strategic reasoning in complex multi-agent environments beyond dialogue. Second, while our datasets cover foundational interaction patterns, exploring additional scenarios would further validate the generality of our approach, such as negotiation (balancing cooperation and competition) (Hua et al., 2024), and multi-party interactions (Ki et al., 2025) (involving mixed motives among multiple agents). Finally, investigating the interplay between strategic reasoning and other cognitive capabilities (e.g., long-term planning (Wang et al., 2025a; Song et al., 2023), theory of mind (Xiao et al., 2025), mental support (Wang et al., 2024)) presents an exciting research direction.

References

M. Abdulhai and 1 others. 2024. Guess my city dataset. Accessed: 2025-09-24.

AI@Meta. 2024. Llama 3 model card. https://huggingface.co/meta-llama/Meta-Llama-3-8B. Accessed: 2025-09-23.

Akinator. 2007. Akinator. https://en.akinator.com. Accessed: 2025-09-24.

Shengnan An, Zeqi Lin, Qiang Fu, Bei Chen, Nanning Zheng, Jian-Guang Lou, and Dongmei Zhang. 2023. How do in-context examples affect compositional generalization? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11027–11052.

Pengyu Cheng, Yong Dai, Tianhao Hu, Han Xu, Zhisong Zhang, Lei Han, Nan Du, and Xiaolong Li. 2024. Self-playing adversarial language game enhances llm reasoning. *Advances in Neural Information Processing Systems*, 37:126515–126543.

CLiPS. 2023. 20q dataset. Accessed: 2025-09-24.

Wenhao Ding, Haohong Lin, Bo Li, and Ding Zhao. 2022. Generalizing goal-conditioned reinforcement learning with variational causal reasoning. *Advances in Neural Information Processing Systems*, 35:26532–26548.

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. *Advances in Neural Information Processing Systems*, 37:28219–28253.

Xidong Feng, Yicheng Luo, Ziyan Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Mguni, Yali Du, and Jun Wang. 2023. Chessgpt: Bridging policy learning and language modeling. *Advances in Neural Information Processing Systems*, 36:7216–7262.

Jakob Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. 2018. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 122–130.

Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998.

Kanishk Gandhi, Dorsa Sadigh, and Noah Goodman. 2023. Strategic reasoning with language models. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Yuncheng Hua, Lizhen Qu, and Reza Haf. 2024. Assistive large language model agents for socially-aware negotiation dialogues. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8047–8074, Miami, Florida, USA. Association for Computational Linguistics.

Biwei Huang, Fan Feng, Chaochao Lu, Sara Magliacane, and Kun Zhang. 2022. Adarl: What, where, and how to adapt in transfer reinforcement learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Chenghao Huang, Yanbo Cao, Yinlong Wen, Tao Zhou, and Yanru Zhang. 2024. Pokergpt: An end-to-end lightweight solver for multi-player texas hold'em via large language model. *ArXiv preprint*, abs/2401.06781.

Jen-tse Huang, Eric John Li, Man Ho Lam, Tian Liang, Wenxuan Wang, Youliang Yuan, Wenxiang Jiao, Xing Wang, Zhaopeng Tu, and Michael R. Lyu. 2025. Competing large language models in multi-agent gaming environments. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*.

Hyeonbin Hwang, Doyoung Kim, Seungone Kim, Seonghyeon Ye, and Minjoon Seo. 2024. Self-explore: Enhancing mathematical reasoning in language models with fine-grained rewards. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1444–1466, Miami, Florida, USA. Association for Computational Linguistics.

Arnav Kumar Jain, Gonzalo Gonzalez-Pumariega, Wayne Chen, Alexander M Rush, Wenting Zhao, and Sanjiban Choudhury. 2025. Multi-turn code generation through single-step rewards. In *Forty-second International Conference on Machine Learning*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Dayeon Ki, Rachel Rudinger, Tianyi Zhou, and Marine Carpuat. 2025. Multiple LLM agents debate for equitable cultural alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24841–24877, Vienna, Austria. Association for Computational Linguistics.

Yihuai Lan, Zhiqiang Hu, Lei Wang, Yang Wang, Deheng Ye, Peilin Zhao, Ee-Peng Lim, Hui Xiong, and Hao Wang. 2024. Llm-based agent society investigation: Collaboration and confrontation in avalon gameplay. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 128–145.

Jonathan Light, Min Cai, Sheng Shen, and Ziniu Hu. 2023. Avalonbench: Evaluating llms playing the game of avalon. *ArXiv preprint*, abs/2310.05036.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Gabriel Mukobi, Hannah Erlebach, Niklas Lauffer, Lewis Hammond, Alan Chan, and Jesse Clifton. 2023. Welfare diplomacy: Benchmarking language model cooperation. In *Socially Responsible Language Modelling Research*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824.

Manish Prajapat, Kamyar Azizzadenesheli, Alexander Liniger, Yisong Yue, and Anima Anandkumar. 2021. Competitive policy optimization. In *Uncertainty in Artificial Intelligence*, pages 64–74. PMLR.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *ArXiv preprint*, abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan 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.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M. Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Hanlin Wang, Chak Tou Leong, Jiashuo Wang, Jian Wang, and Wenjie Li. 2025a. Spa-rl: Reinforcing llm agents via stepwise progress attribution. *Preprint*, arXiv:2505.20732.

Jiashuo Wang, Yi Cheng, and Wenjie Li. 2022. CARE: Causality reasoning for empathetic responses by conditional graph generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 729–741, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiashuo Wang, Chunpu Xu, Chak Tou Leong, Wenjie Li, and Jing Li. 2024. Muffin: Mitigating unhelpfulness in emotional support conversations with multifaceted AI feedback. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 567–585, Bangkok, Thailand. Association for Computational Linguistics.

Shenzhi Wang, Chang Liu, Zilong Zheng, Siyuan Qi, Shuo Chen, Qisen Yang, Andrew Zhao, Chaofei Wang, Shiji Song, and Gao Huang. 2023. Avalon's game of thoughts: Battle against deception through recursive contemplation. *ArXiv preprint*, abs/2310.01320.

Siyuan Wang, Enda Zhao, and Xiang Ren. 2025b. Stepwise informativeness search for improving LLM reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25291–25309, Suzhou, China. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yang Xiao, Jiashuo Wang, Qiancheng Xu, Changhe Song, Chunpu Xu, Yi Cheng, Wenjie Li, and Pengfei Liu. 2025. Towards dynamic theory of mind: Evaluating LLM adaptation to temporal evolution of human states. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24036–24057, Vienna, Austria. Association for Computational Linguistics.

Qiancheng Xu, Yongqi Li, Heming Xia, Fan Liu, Min Yang, and Wenjie Li. 2025. PEToolLLM: Towards personalized tool learning in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21488–21503, Vienna, Austria. Association for Computational Linguistics.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. 2024. Exploring large language models for communication games: An empirical study on werewolf. *Preprint*, arXiv:2309.04658.

Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun MA, and Bo An. 2025. Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. In *NeurIPS 2025 Fourth Workshop on Deep Learning for Code*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Yuan Yao, Haoxi Zhong, Zhengyan Zhang, Xu Han, Xiaozhi Wang, Kai Zhang, Chaojun Xiao, Guoyang Zeng, Zhiyuan Liu, and Maosong Sun. 2021. Adversarial language games for advanced natural language intelligence. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence,* *EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14248–14256. AAAI Press.

Arianna Yuan, Will Monroe, Yue Bai, and Nate Kushman. 2018. Understanding the rational speech act model. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 40.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772.

Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, Zeqi Tan, Peng Li, Yueting Zhuang, and Weiming Lu. 2024b. Agent-pro: Learning to evolve via policy-level reflection and optimization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5348–5375, Bangkok, Thailand. Association for Computational Linguistics.

Yadong Zhang, Shaoguang Mao, Tao Ge, Xun Wang, Yan Xia, Wenshan Wu, Ting Song, Man Lan, and Furu Wei. 2024c. Llm as a mastermind: A survey of strategic reasoning with large language models. In *First Conference on Language Modeling*.

Yizhe Zhang, Jiarui Lu, and Navdeep Jaitly. 2024d. Probing the multi-turn planning capabilities of LLMs via 20 question games. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1495–1516, Bangkok, Thailand. Association for Computational Linguistics.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. 2024. Archer: Training language model agents via hierarchical multi-turn rl. In *International Conference on Machine Learning*, pages 62178–62209. PMLR.

Appendix Contents

A FoPO Derivation 13

B Cooperative RSA 14
- B.1 Game Rules 14
- B.2 Bayesian Inference 14
- B.3 Example in Figure 2 15
- B.4 Speaker & Listener Foresight . . . 16
- B.5 Data Construction 16
- B.6 Game Reward 17

C Experiment 17
- C.1 Comparison Models 17
- C.2 Training Datasets 18
- C.3 Training Details 18
- C.4 Code Details 19
- C.5 GRPO Fails on Cooperative RSA . 19
- C.6 Out-of-Domain Evaluation 19

D Case Study 19
- D.1 Dataset Cases 19
- D.2 FoPO Generation Cases 19

E LLM Agent Prompts 20
- E.1 Task Prompts 20
- E.2 Role Prompts 21

A FoPO Derivation

The gain of agent 1 can be represented by the surrogate objective $\mathcal{O}^1(\text{Agent 1}, \text{Agent 2})$, which depends on both agents' strategies. A core idea in FoPO is that agent 1 should take an action that not only maximizes its final reward based on the current generated action, but also anticipates and responds to how agent 2 might change its behavior after its own optimization. Following opponent learning, we aim to maximize the "foresight gain":

$$\mathcal{O}^1\left(\pi(\text{prompt}^1, \theta),\ \pi(\text{prompt}^2, \theta + \Delta\theta)\right), \quad (11)$$

where $\Delta\theta$ is the direction in which agent 2 updates its policy during optimization.

Assuming $\Delta\theta$ is small, we can apply a first-order Taylor expansion with respect to the second argument:

$$\mathcal{O}^1(\theta,\ \theta + \Delta\theta) \approx \mathcal{O}^1(\theta, \theta) + (\Delta\theta)^\top \nabla_{\theta_2} \mathcal{O}^1(\theta, \theta_2)\Big|_{\theta_2=\theta}, \quad (12)$$

where the notation $\mathcal{O}^1(\theta_1, \theta_2)$ represents agent 1's value when the two agents use policy parameters θ_1 and θ_2, respectively. The agent 2's update, given the foresight weight η, is:

$$\Delta\theta = \eta \nabla_{\theta_2} \mathcal{O}^2(\theta, \theta_2)\Big|_{\theta_2=\theta}, \quad (13)$$

Substituting this into the Taylor expansion and differentiating $\mathcal{O}^1(\theta, \theta + \Delta\theta)$ with respect to θ, we obtain the FoPO update rule:

$$\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_\theta \mathcal{O}^1(\theta, \theta) + \alpha\eta \left(\nabla_{\theta_2} \mathcal{O}^1(\theta, \theta_2)\right)^\top \nabla_\theta \nabla_{\theta_2} \mathcal{O}^2(\theta, \theta_2)\Big|_{\theta_2=\theta}. \quad (14)$$

Unlike prior opponent modeling approaches (Foerster et al., 2018; Prajapat et al., 2021), we deliberately truncate the dependency of $\nabla_{\theta_2}\mathcal{O}^1(\theta, \theta_2)$ on θ_1. This design choice allows us to focus on the forward influence: how self's current behavior shapes the counterpart's subsequent policy update, and how this adapted policy then affects self's future returns. Instead of differentiating through the opponent's learning rule, we compute this influence via the forward chain rule, avoiding the need to backpropagate through the counterpart's gradient computation and thus eliminating the requirement for mixed Hessian terms $\frac{\partial^2 \mathcal{O}^1}{\partial\theta_1 \partial\theta_2}$. This approach sidesteps the prohibitive computational costs and complexity that characterize previous opponent modeling methods, making it well-suited for enhancing the strategic reasoning of LLM-based agents.

Table 3 provides the interpretation of each term in Equation (14). Finally, the FoPO updates the parameters by

$$\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_\theta r_t^1(\theta) \hat{A}_t^{1,\text{clip}} + \alpha\eta \underbrace{\left[r_t^1(\theta) \hat{A}_t^{1,\text{clip}} \nabla_\theta r_{t+1}^2(\theta)\right]^\top}_{\text{Sensitivity of } \mathcal{O}^1 \text{ to } 2} \underbrace{\nabla_\theta r_t^1(\theta) \nabla_\theta r_{t+1}^2(\theta) \hat{A}_{t+1}^{2,\text{clip}}}_{\text{Effect of 1 on } \mathcal{O}^2}. \quad (15)$$

When the KL divergence term is included, the formulation becomes equivalent to Equation (7). Although FoPO is applicable to agents with differing parameters, our approach focuses on enhancing LLMs' strategic reasoning via self-play.

Term	Representation	Interpretation
$\nabla_\theta \mathcal{O}^1(\theta, \theta)$	$\nabla_\theta r_t^1(\theta) \hat{A}_t^{1,\text{clip}}$	The standard policy gradient of agent 1, representing local improvement of its own return.
$\nabla_{\theta_2} \mathcal{O}^1(\theta, \theta_2)$	$\nabla_{\theta_2} \left(r_t^1(\theta) r_{t+1}^2(\theta) \hat{A}_t^{1,\text{clip}} \right)$ $= r_t^1(\theta) \nabla_{\theta_2} r_{t+1}^2(\theta) \hat{A}_t^{1,\text{clip}}$ $\xrightarrow{\theta_2=\theta} r_t^1(\theta) \nabla_\theta r_{t+1}^2(\theta) \hat{A}_t^{1,\text{clip}}$	Measures how agent 1's gain changes in response to updates in agent 2's policy parameters.
$\nabla_\theta \nabla_{\theta_2} \mathcal{O}^2(\theta, \theta_2)$	$\nabla_\theta \nabla_{\theta_2} \left[r_t^1(\theta) r_{t+1}^2(\theta) \hat{A}_{t+1}^{(2,\text{clip}} \right]$ $= \nabla_\theta r_t^1(\theta) \nabla_{\theta_2} r_{t+1}^2(\theta) \hat{A}_{t+1}^{(2,\text{clip}}$ $\xrightarrow{\theta_2=\theta} \nabla_\theta r_t^1(\theta) \nabla_\theta r_{t+1}^2(\theta) \hat{A}_{t+1}^{(2,\text{clip}}$	Reflects how agent 2's learning dynamics are influenced by agent 1's policy.

Table 3: Interpretation of each term in the FoPO update.

B Cooperative RSA

B.1 Game Rules

The Cooperative RSA game is a multi-turn interaction between a speaker and a listener. Both agents share an object list $O = \{o_1, \ldots, o_N\}$, where each object has M binary-valued features $F = \{f_1, f_2, \ldots, f_M\}$. The speaker refers to a target object $\hat{o}$ by revealing one feature per turn, while the listener responds with a subset of objects consistent with the received feature. The game succeeds if the listener isolates $\hat{o}$ as a singleton set and fails if the target is ever excluded. This setup encourages pragmatic reasoning: the speaker must select informative features strategically, and the listener incrementally refines its hypotheses. More efficient interactions, requiring fewer turns, reflect stronger alignment and reasoning capabilities.

B.2 Bayesian Inference

The behaviors of rational speakers and listeners are modeled via a Bayesian process. We divide the interaction at the t-th and $(t+1)$-th turns into the speaker and listener sides.

Speaker At turn t, the speaker evaluates each feature $f_m(\hat{o})$ of the target object $\hat{o}$ given candidate objects $O^{(t)}$. The literal speaker assumes a uniform prior over objects containing the specific feature $f_m(\hat{o})$:

$$P_{L_1}(o_n \mid \hat{f}^{(t)}, O) = \begin{cases} |\hat{f}^{(t)}|^{-1}, & o_n \in C, \\ 0, & \text{otherwise.} \end{cases}$$

The rational speaker applies Bayesian inference to calculate the posterior for feature $f_m(\hat{o})$

$$P_{L_0}(o_n \mid f_m(\hat{o}), O^{(t)}) = \frac{P(f_m(\hat{o}) \mid o_n, O^{(t)})\, P(o_n)}{\sum_{o \in O^{(t)}} P(f_m(\hat{o}) \mid o, O^{(t)})\, P(o)},$$

with likelihood:

$$P(f_m(\hat{o}) \mid \hat{o}, O^{(t)}) = \frac{|f_m(\hat{o})|^{-1}}{\sum_{f \in F} |f|^{-1}},$$

where $|f|$ is the number of objects in $O^{(t)}$ possessing feature f.

For each feature, the target rank is

$$\text{rank}_{f_m(\hat{o})}(\hat{o}) = |\{o_n \in O^{(t)} : P_{L_0}(o_n \mid f_m(\hat{o}), O^{(t)}) \geq P_{L_0}(\hat{o} \mid f_m(\hat{o}), O^{(t)})\}|$$

The speaker selects the feature with the highest discriminability:

$$\hat{f}^{(t)} = \arg \min_{m=1,\ldots,M} \text{rank}_{f_m(\hat{o})}(\hat{o}).$$

Listener At turn $(t+1)$, the listener observes $\hat{f}^{(t)}$ and updates its posterior:

$$P_{L_1}(o_n \mid \hat{f}^{(t)}, O^{(t)}) = \frac{P(\hat{f}^{(t)} \mid o_n, O^{(t)})\, P(o_n)}{\sum_{o \in O^{(t)}} P(\hat{f}^{(t)} \mid o, O^{(t)})\, P(o)},$$

with the same likelihood as above.

To model pragmatic inference, the listener simulates the speaker's choice:

1. For each $o_n \in O^{(t)}$ with $\hat{f}^{(t)} \in o_n$, compute features $F(o_n)$.
2. Simulate the speaker selecting the most informative feature:

$$f^*_{o_n} = \arg\max_{f \in F(o_n)} P_{L_0}(o_n \mid f, O^{(t)}).$$

3. Retain o_n if $f^*_{o_n} = \hat{f}^{(t)}$.

The listener's belief set BeliefSet$(\hat{f}^{(t)})$ is formulated as

$$\left\{ o_n \in O^{(t)} \;\middle|\; \hat{f}^{(t)} \in o_n \text{ and } f^*_{o_n} = \hat{f}^{(t)} \right\}.$$

The next candidate set is $O^{(t+2)}$ is

$$\arg\max_{o_n \in \text{BeliefSet}(\hat{f}^{(t)})} P_{L_1}(o_n \mid \hat{f}^{(t)}, O^{(t)}).$$

If only one object remains, it is returned as the final selection.

B.3 Example in Figure 2

Consider the example in Figure 2, where the object set is O = {dry-blue-smooth-square, wet-green-rough-square, wet-green-smooth-square, wet-blue-smooth-circle, wet-blue-smooth-square, dry-blue-rough-circle, dry-blue-rough-square, dry-blue-smooth-circle}, with target object $\hat{o} = o_8 =$ dry-blue-smooth-circle. Let the features be moisture, color, texture, and shape.

Speaker Calculation First, The speaker evaluates each feature $f_m(\hat{o})$ to choose the most informative one:

(1) Moisture = dry: occurs in {dry-blue-smooth-square, dry-blue-rough-circle, dry-blue-rough-square, dry-blue-smooth-circle}.

$$P(\text{dry} \mid o_1, O) = \frac{\frac{1}{4}}{\frac{1}{4}+\frac{1}{6}+\frac{1}{5}+\frac{1}{5}} = \frac{15}{49}$$

$$P(\text{dry} \mid o_6, O) = \frac{\frac{1}{4}}{\frac{1}{4}+\frac{1}{6}+\frac{1}{3}+\frac{1}{3}} = \frac{3}{13}$$

$$P(\text{dry} \mid o_7, O) = \frac{\frac{1}{4}}{\frac{1}{4}+\frac{1}{6}+\frac{1}{3}+\frac{1}{5}} = \frac{15}{57}$$

$$P(\text{dry} \mid o_8, O) = \frac{\frac{1}{4}}{\frac{1}{4}+\frac{1}{6}+\frac{1}{3}+\frac{1}{5}} = \frac{15}{57}$$

$$P(\hat{o_1} \mid \text{dry}, O) = \frac{\frac{15}{49}}{\frac{15}{49}+\frac{3}{13}+\frac{15}{57}+\frac{15}{57}}$$

$$P(\hat{o_6} \mid \text{dry}, O) = \frac{\frac{3}{13}}{\frac{15}{49}+\frac{3}{13}+\frac{15}{57}+\frac{15}{57}}$$

$$P(\hat{o_7} \mid \text{dry}, O) = \frac{\frac{15}{57}}{\frac{15}{49}+\frac{3}{13}+\frac{15}{57}+\frac{15}{57}}$$

$$P(\hat{o_8} \mid \text{dry}, O) = \frac{\frac{15}{57}}{\frac{15}{49}+\frac{3}{13}+\frac{15}{57}+\frac{15}{57}}$$

The rank of the target object would be the second, sharing the spot with one other object (dry-blue -rough-square).

(2) Color = blue: occurs in {dry-blue-smooth -square, wet-blue-smooth-circle, wet-blue-smooth -square, dry-blue-rough-circle, dry-blue-rough -square, dry-blue-smooth-circle}. The ranking of the target object is the third, sharing the spot with one other object (wet-blue-smooth-circle).

(3) Texture = smooth: occurs in {dry-blue -smooth-square, wet-green-smooth-square, wet -blue-smooth-circle, wet-blue-smooth-square, dry -blue-smooth-circle}. The ranking of the target object is third, sharing the spot with one other object (wet-blue-smooth-circle).

(4) Shape = circle. occurs in {dry-blue-smooth-square, wet-green-rough-square, wet-green-smooth-square, wet-blue-smooth-square, dry-blue-rough-square}. The ranking of the target object is one of the first, sharing the spot with one other object (wet-blue-smooth-circle).

By comparing the above rankings, a rational speaker would select the most informative feature, which is "Shape=circle," in the first turn.

Listener Calculation At the second turn, the listener observes $\hat{f}^{(t)}$ ="Shape=circle." First, compute the prior (literal posterior) for each object:

$$P_{L_1}(o_n \mid \hat{f}^{(t)}, O) = \begin{cases} |\hat{f}^{(t)}|^{-1}, & o_n \in C, \\ 0, & \text{otherwise.} \end{cases}$$

$$C = \{ \text{dry-blue-smooth-circle}, \text{wet-blue-smooth-circle}, \text{dry-blue-rough-circle} \}$$

A rational listener simulates the speaker's choice among each candidate in {dry-blue-smooth-circle, wet-blue-smooth-circle, dry-blue-rough-circle} is as follows:

For the first object **(1) dry-blue-smooth-circle**: If the speaker wants to refer to the object dry-blue-smooth-circle, it is most likely that the speaker would say "circle." "Circle" matches the speaker's utterance. Therefore, dry-blue-smooth-circle is retained in BeliefSet($\widehat{\text{dry}}^{(t)}$). Refer to the previous section for the calculation process. **(2) wet-blue-smooth-circle:** Similar to the previous calculation, to refer to this object, the speaker should have said "circle." "Circle" matches the speaker's utterance. Therefore, wet-blue-smooth-circle is retained in BeliefSet($\widehat{\text{dry}}^{(t)}$). **(3) dry-blue-rough-circle:** Similar to the previous calculation, to refer to this object, the speaker should have said "rough." "Rough" does not match the speaker's utterance. Therefore, dry-blue-rough-circle is not retained in BeliefSet($\widehat{\text{dry}}^{(t)}$). Then, the candidate set is

$$\begin{aligned} O^{(t+2)} &= \text{BeliefSet}(\widehat{\text{circle}}^{(t)}) \\ &= \{\text{dry-blue-smooth-circle}, \\ &\quad \text{wet-blue-smooth-circle}\}. \end{aligned} \tag{16}$$

B.4 Speaker & Listener Foresight

In the cooperative RSA game, the speaker's foresight can be naturally modeled via Bayesian inference as formulated in RSA, whereas the listener's foresight is more complex. As described in Appendix B.2, a rational listener reasons based solely on the speaker's behavior in the previous turn, implicitly assuming the speaker is rational. In practice, however, the listener cannot be certain of the speaker's rationality and must therefore account for multiple possible courses of the speaker's future actions. This process is illustrated in Figure 6.

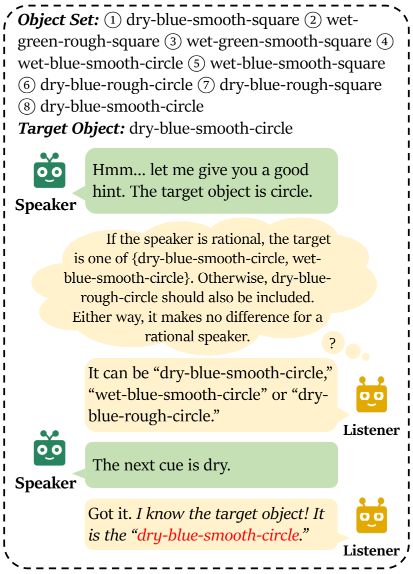

Figure 6: PPO optimized the self policy (π_1) in isolation, while FoPO introduces foresight into the future updates of the counterpart Policy (π_2).

B.5 Data Construction

The data construction pipeline can be summarized as: (Feature Pair Bank Construction, Objective Matrix and Object Construction) → Dialogue Chain Computation → LLM-based Dialogue Generation. Each step is illustrated as follows:

Feature Pair Bank Construction We first construct a curated set of binary feature pairs, each representing a minimal semantic contrast (e.g., *smooth* vs. *rough*, *graceful* vs. *clunky*). This bank is partitioned into two disjoint subsets to prevent data leakage between training stages.

- **SFT Feature Pair Bank:** A set of 86 pairs used to generate polished dialogue for supervised fine-tuning.
- **RL Feature Pair Bank:** A set of 25 pairs used exclusively to construct ranking-based preference data for reinforcement learning.

This separation ensures a clean experimental boundary between learning phases, as the RL component does not optimize on features the model has already seen during supervised training.

Objective Matrix and Object Construction Each matrix in our system encodes a semantic mapping between feature dimensions and a set of candidate referents. An entry of 1 indicates that a referent shares the same value as the target referent for a given feature, while 0 denotes a mismatch.

$$M_{i,j} = \begin{cases} 1, & \text{referent } i \text{ matches target on feature } j \\ 0, & \text{otherwise} \end{cases}$$

We generate a large pool of such binary matrices with varying shapes, denoted as $m \times n$, where n is the number of candidate referents and m is the number of features. To evaluate the reasoning depth required to resolve each matrix, we simulate golden dialogues using Rational Speaker and Listener models. This allows us to annotate each matrix with the number of rounds required to uniquely identify the target referent through pragmatic inference. These selected matrices, along with features from the feature pair bank, are then used to construct the object list and specify the target object for each dialogue task.

Dialogue Chain Computation Using the constructed object list and the target object, we employ the RSA model, illustrated in Appendix B.2, to compute the optimal dialogue chain. This process involves iterative pragmatic inference, where a rational speaker chooses an utterance that maximally reduces the listener's uncertainty about the target object, and a rational listener updates their belief distribution accordingly. The output is a sequence of features and object sets updates representing the most efficient path to identifying the target.

LLM-based Dialogue Generation The final step is to use an LLM to translate the computed dialogue chain into a natural, conversational format. The LLM takes the structured output of the Bayesian computation as input and generates a realistic dialogue that mirrors the pragmatic choices and reasoning depth of the chain, thereby creating a rich dataset for training and evaluation. In this process, we employed four prompts, with one representative example shown below:

```
Representative RSA Conversation Generation Prompt:

You're awesome at making dialogue sound natural
and conversational! I need your help turning this
robotic dialogue into something that feels like real
people chatting.

Scenario Overview:
- This is a guessing game: the Speaker describes an
object, and the Listener tries to guess what it is.
- The target object the Speaker is referring to is:
{target_referent}.
- The Listener needs to figure out what object the Speaker
means, using this format when they finally guess: "I
know the target object. It is ..."
- Here are all the possible objects being referred to:
{referent_set}.

Original dialogue:
{dialogue}

This dialogue serves as the backbone of your
refined version. Your task is to revise it to a
real-world conversation, while maintaining the basic
contents: the feature or the object(s).

Transform the original dialogue to sound friendly,
casual, and human, while keeping the structure and
meaning the same. Instructions for the generated
dialogue:
1. Keep the same number of lines, turns, and speakers
as the original.
2. Each casual line must match the original's meaning
and content, just in a more natural tone.
3. Make it sound like real people chatting-relaxed,
informal, and friendly.
4. Use casual phrases, natural pauses, filler words
(like "um," "you know"), and everyday language.
5. Keep each line around 70 words-brief, but with a
conversational feel.

Output Format:
Just give me the improved dialogue in this exact format:
Speaker: [Casual version]
Listener: [Casual version]
Speaker: [Casual version]
Listener: [Casual version]
...
```

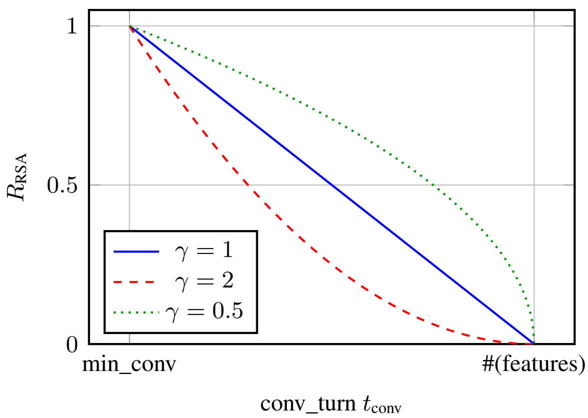

Figure 7: Higher γ leads to stronger penalties for exceeding optimal turns.

B.6 Game Reward

The reward in Cooperative RSA is strongly affected by the parameter γ, as shown in Figure 7. Values of $\gamma > 1$ strongly encourage agents to complete the game in fewer turns, whereas $\gamma < 1$ offers a more moderate incentive.

C Experiment

C.1 Comparison Models

To prove the effectiveness of FoPO, we consider the following comparison methods: (1) **ICT** (An et al., 2023): In context learning augments each training instance with k in-context demonstrations and trains with next-token cross-entropy, distilling demonstration patterns into the model parameters.

Backbone	Training Set	Method	Guessing	Bar	Dollar	Diner	Auction	Battle	Pirate	Avg.
Llama-3-8B-Instruct	20 Questions	*SFT*	76.25	63.67	**64.99**	85.50	20.49	19.59	55.81	55.19
		PPO	78.67	64.35	64.89	87.34	27.89	**13.18**	57.30	56.23
		FoPO	**79.59**	**65.00**	64.81	**91.00**	**28.24**	11.62	**58.26**	**56.93**
	Guess My City	*SFT*	90.28	53.17	53.28	91.30	26.17	15.54	43.88	53.37
		PPO	**90.89**	55.83	53.39	91.50	**26.18**	14.19	42.58	53.51
		FoPO	89.79	**55.92**	**55.37**	**91.71**	26.13	**17.65**	**45.67**	**54.61**
Qwen3-14B	20 Questions	*SFT*	93.66	**36.83**	**77.79**	6.40	**10.93**	78.14	81.88	55.09
		PPO	93.66	35.41	74.13	8.00	10.10	83.11	85.33	55.68
		FoPO	**94.59**	33.50	77.86	**9.40**	10.87	**86.90**	**92.17**	**57.90**
	Guess My City	*SFT*	94.60	36.67	71.55	9.10	11.04	80.00	83.70	55.24
		PPO	**94.70**	38.48	77.18	8.70	10.69	**82.32**	83.17	56.46
		FoPO	94.69	**42.00**	**94.52**	**9.20**	**11.24**	76.58	**84.07**	**58.90**

Table 4: PPO and FoPO Performance on γ-Bench with deductive reasoning training

Backbone	Guessing	Bar	Dollar	Diner	Auction	Battle	Pirate	Avg.
GPT-4.1	95.15	35.83	93.12	24.80	13.24	35.00	94.92	56.01
DeepSeek-V3.2	94.83	26.67	96.65	5.72	13.26	64.68	99.90	57.39

Table 5: GPT-4.1 and DeepSeek-V3.2 Performance on γ-Bench

This baseline is included to assess whether the observed improvements stem from learning from rewards or from conversation patterns in the training data. We include this baseline to prove that SFT models are fully trained. (2) **PPO** (Schulman et al., 2017): We include PPO since our FoPO method builds upon it, allowing us to isolate the effect of the foresight-oriented correction in FoPO. (3) **ArCHer** (Zhou et al., 2024): This is a hierarchical RL algorithm, where a high-level RL algorithm is used to train a value function that aggregates rewards over entire utterances and a low-level RL algorithm then leverages this high-level value function to train a token-by-token policy. Due to the high-level RL, the model can plan across utterances and guide the low-level policy with broader conversational objectives. We include it to compare its explicit long-term planning capability with the explicit counterpart foresight offered by FoPO. (4) **GRPO** (Shao et al., 2024): We further apply the foresight optimization on top of GRPO to demonstrate the generalizability of our proposed method.

C.2 Training Datasets

When fine-tuning models on 20 Questions and GuessMyCity, we randomly sampled 2,300 conversations from each dataset to ensure a comparable number of training steps, and additional 240 instances (keywords or cities) for further RL in both tasks for the same purpose.

C.3 Training Details

To improve training efficiency, we applied LoRA (Hu et al., 2022) during SFT and subsequently merged the LoRA parameters into the backbone model. We set the rank as 8, alpha as 16, and applied the LoRA modules to the query and value projection layers. We trained the models using the AdamW optimizer. For SFT, we set the learning rate α to 5×10^{-5}, the KL regularization coefficient β to 0.01, and the batch size to 32. For RL and ICT, the learning rate α was 1×10^{-5}, β was 0.1, η was 0.1, the reward decay factor δ is 0.8, and the batch size was 16. Training was performed on 4 NVIDIA 5880 (48GB) GPUs. We employ DeepSpeed ZeRO Stage 2 (Rajbhandari et al., 2020) to optimize memory usage and accelerate training. For GRPO, we sample four trajectories for each instance. For PPO and FoPO, we follow the GRPO training protocol and do not train a critic model. This design choice enhances training efficiency and facilitates a more equitable comparison across various policy optimization methods.

To determine the most suitable value for η, we evaluated several candidates using a smaller RL training dataset and Meta-Llama-3-8B-Instruct SFT models. Specifically, we trained different FoPO variants with varying η values. With the SFT model serving as the counterpart policy, we evaluated each variant via reward on cooperative RSA and win rate on competitive Taboo. As shown

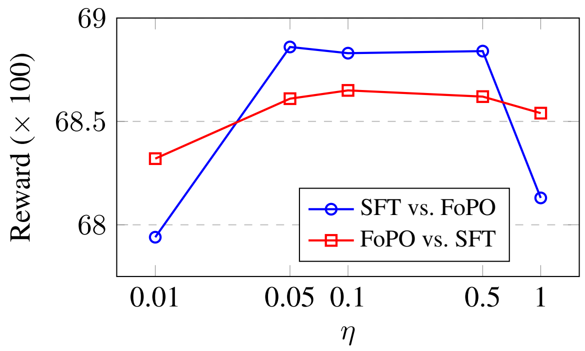

Figure 8: Hyperparameter sensitivity of FoPO to the foresight weight η on cooperative RSA (reward). Performance is relatively stable across $\eta \in [0.05, 0.5]$.

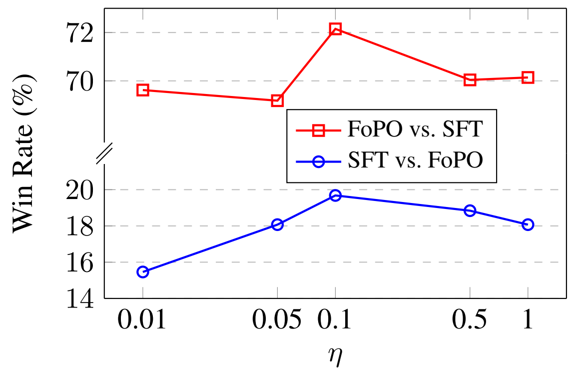

Figure 9: Hyperparameter sensitivity of FoPO to the foresight weight η on competitive Taboo (win rate). $\eta = 0.1$ achieves the best performance.

in Figure 8 and Figure 9, $\eta = 0.1$ achieves the best performance across both tasks and was therefore adopted in all experiments.

C.4 Code Details

To implement FoPO, the typical `backward()` pass is replaced by four separate calls to `torch.autograd.grad` to compute and assign gradients for each parameter via `p.grad = ...`. By clearing the computational graph's cache after each call, we ensure that GPU memory usage remains on par with a standard PPO implementation. FoPO's total training time is roughly 3–3.5× that of PPO due to the multiple gradient computations. Compared to PPO, the training time for ArCHer is about eight times greater. Its memory usage also fluctuates, with a peak consumption that is approximately twice PPO's. The datasets themselves will be released upon acceptance of this work.

C.5 GRPO Fails on Cooperative RSA

Since we observed that GRPO fails on Cooperative RSA, we investigate whether removing standard deviation normalization, following DR.GRPO (Liu et al., 2025), could mitigate the issue. However, DR.GRPO exhibits identical failure modes: models still collapse to deterministically outputting a single token (probability 1.0) at all positions. This indicates the core problem lies in the mean-based advantage computation, which becomes unreliable given the vast dialogue space and limited sampling.

C.6 Out-of-Domain Evaluation

We evaluate FoPO on deductive reasoning tasks. Specifically, PPO and FoPO were trained on 20 Questions and Guess My City datasets, and their performance was assessed using γ-Bench. The results are presented in Table 4. The results indicate that, although FoPO is primarily designed for strategic reasoning, it can also enhance model performance when trained on deductive reasoning tasks. This highlights the ability of foresight optimization to generalize across multiple task types.

For reference, Table 5 reports the performance of GPT-4.1 and DeepSeek-V3.2, the closed-source models used for dataset construction. Their results are comparable to or slightly below those of the SFT models, yet consistently higher than those of the vanilla models (Table 1). This suggests that the structured rules in our data collection pipeline improve dataset quality beyond what purely distilled data would yield.

D Case Study

D.1 Dataset Cases

Table 6 presents a sample from the Cooperative RSA dataset, including the conversation chain, with a minimum of three rounds in the conversation. Table 7 presents a sample from the Competitive Taboo dataset, where the target word is "one" and the winner is the defender.

D.2 FoPO Generation Cases

The generated conversations from the Cooperative RSA and Competitive Taboo tasks, produced by FoPO Llama-3-8B-Instruct, are shown in Table 8 and Table 9. In the Cooperative RSA task, the speaker and listener succeed within the minimum conversation turns. In the Competitive Taboo task, the defender won by successfully guessing the target word.

Target Object: loud-weak-late-lean	
Object List: loud-weak-late-lean, quiet-strong-early-fat, quiet-strong-early-lean, quiet-strong-late-fat, quiet-strong-late-lean, loud-weak-early-fat, loud-weak-early-lean, loud-weak-late-fat	
Chain: [Speaker: loud, Listener: loud-weak-late-lean, loud-weak-early-fat, loud-weak-early-lean, loud-weak-late-fat, Speaker: late, Listener: loud-weak-late-lean, loud-weak-late-fat, Speaker: lean, Listener: loud-weak-late-lean]	
Conversation	
Role	**Turn**
Speaker	To initiate the identification process, I would like to highlight the attribute "loud" as a salient feature of the target object. Please consider all candidate objects that possess this particular characteristic as we proceed with the elimination and selection process.
Listener	In response to your specification, I have narrowed the potential candidates to the following objects, each of which is characterized by the attribute "loud": ('loud, weak, late, lean'), ('loud, weak, early, fat'), ('loud, weak, early, lean'), and ('loud, weak, late, fat').
Speaker	Building upon the previous description, I would now like to specify that the target object is further distinguished by the attribute "late." Kindly refine your consideration to those objects which simultaneously exhibit both "loud" and "late" as defining features.
Listener	Taking into account your most recent clarification, I have reduced the set of plausible candidates to two objects: ('loud, weak, late, lean') and ('loud, weak, late, fat'), both of which possess the attributes "loud" and "late."
Speaker	To further delineate the target object, I would like to introduce the attribute "lean" as a critical distinguishing characteristic. This should enable you to identify the precise object among the remaining options.
Listener	Based on the cumulative information you have provided—specifically, the attributes "loud," "late," and "lean"—I am confident in my assessment. I know the target object. It is ('loud, weak, late, lean').

Table 6: A data instance from the Cooperative RSA dataset.

Target Word: One	
Conversation	
Role	**Turn**
Attacker	If you had to choose a number that symbolizes unity or the beginning of a sequence, which would you pick?
Defender	I would probably select something straightforward, perhaps the initial digit used when counting.
Attacker	That makes sense! Often, when people start counting out loud, what's the first thing they typically say?
Defender	When people start counting out loud, they usually say the very first numeral, often used as the starting point for sequences.
Attacker	Consider numbers that are often viewed as singular or representing individuality—what digit comes to mind first?
Defender	I know the word! It is "one!"

Table 7: A data instance from the Competitive Taboo dataset.

E LLM Agent Prompts

E.1 Task Prompts

For each task, we prepare a set of prompts, from which a single prompt is randomly selected during training. Notably, the prompts for Competitive Taboo are also used to generate conversations for the corresponding dataset. We provide several representative prompts here, while the full set is included in the code.

For the Cooperative RSA task, the following prompt was among those used in the implementation:

Target Object: muted-rough-soggy-no-bumpy	
Object List: muted-rough-soggy-no-bumpy, echoing-smooth-crisp-yes-flat, echoing-smooth-crisp-no-bumpy, echoing-smooth-soggy-yes-bumpy, muted-rough-crisp-no-bumpy	
Conversation	
Role	**Turn**
Speaker	I'm thinking of something that feels rough to the touch.
Listener	Okay, so that could be either the one that's muted, rough, soggy, and no, or the one that's muted, rough, crisp, and no, but bumpy.
Speaker	And to narrow it down, it's definitely soggy.
Listener	I know the target object. It is muted, rough, soggy, and no, bumpy.

Table 8: FoPO (Llama-3-8B-Instruct) generated Cooperative RSA conversation.

```
Representative Cooperative RSA Task Prompt:

Embark on the collaborative challenge of the
Rational Speech Act Game, where players assume the
roles of either speaker or listener.

The speaker enters the game with a covertly
assigned target object, while the listener starts
without knowledge of this object. The speaker's goal is
to effectively guide the listener toward identifying
the target object, thereby securing victory. However,
there's a rule: the speaker may only provide one
feature per turn.

Simultaneously, the listener's task is to deduce the
target object and present possible target referent
objects at each turn. The listener benefits from the
ability to suggest multiple possible target referent
object sets during their turn. If the listener
identifies the target object, they can declare "I know
```

Target Word: Appetency	
Conversation	
Role	**Turn**
Attacker	Let's begin. Imagine you're at a restaurant, and you're trying to decide what to order. What's the first thing you think about?
Defender	I think about what type of food I'm in the mood for.
Attacker	That's a great point! Now, what if you were at a buffet and you're scanning the options? What's the first thing you consider when choosing what to eat?
Defender	I consider what's visually appealing and what looks fresh.
Attacker	That's a good strategy. Now, let's think about the first stage of the digestive process. What's the term for the desire or craving for food?
Defender	I know the word! It is "appetency."

Table 9: FoPO (Llama-3-8B-Instruct) generated Competitive Taboo conversation, where the FoPO Defender is the winner.

```
the target object! It is 'target object'!"

During each turn, the speaker should carefully choose a feature of the target object that delivers the most valuable information to the listener, while the listener adjusts their possible target referent objects based on the previous turn's information.

Remember, the listener can only update their referent set from the previous turn's guess; they cannot add new referents.

The scoring framework rewards efficiency: the fewer turns required to identify the target object, the higher the score achieved.
```

For Competitive Taboo, the following prompt was used in the implementation:

```
Representative Competitive Taboo Task Prompt:

Play the game of Adversarial Taboo. In this game, there are two players, an attacker and a defender.

At the beginning, the attacker is assigned a target word, with which the defender is not informed. The task of the attacker is to induce the defender to utter the target word unconsciously, and then the attacker wins the game. However, the attacker is not allowed to utter the target word; otherwise, the attacker loses.

At the same time, the defender tries to figure out the target word. If the defender identifies the target word, he can say "I know the word! It is 'target word'!". Then the defender wins. Remember, the defender can only guess the word once; otherwise, he will directly lose the game.

The game has {max_turns} turns. if nobody wins during {max_turns} turns, there is a tie.
```

E.2 Role Prompts

Cooperative RSA. In the task of Cooperative RSA, the prompts for the speaker and the listener are as follows:

```
Speaker Prompt:

### Instruction: You are the pragmatic rational speaker. The target object is '{target}' and the object list is '{object_list}'. Provide your response, including the object feature.

### Response:

Listener Prompt:

### Instruction: You are the pragmatic rational listener. The object list is '{object_list}'. Provide your inferred target object or the possible target object sets.

### Response:
```

Competitive Taboo. In the task of Competitive Taboo, the prompts for the attacker and the defender are as follows:

```
Attacker Prompt:

### Instruction: You are the attacker. The target word is 'target'. Provide your response for the next turn.

### Response:

Defender Prompt:

### Instruction: You are the defender. Provide your response to infer the target word.

### Response:
```