Online Strategic Reasoning for LLM-Based Negotiation via Opponent-Aware Monte Carlo Tree Search

Chen-Cheng Huang[1] **Yi-Cheng Wang**[1] **Hung-Lin Hu**[4]
Yi-Ren Yeh[2] **Chu-Song Chen**[1,3]
[1]Department of Computer Science and Information Engineering, National Taiwan University
[2]Department of Mathematics, National Kaohsiung Normal University
[3]FinTech Center, National Taiwan University
[4]E.SUN Financial Holding Co., Ltd.
[1]{b09902049, d13922033, chusong}@csie.ntu.edu.tw
[4]andy86715@gmail.com [2]yryeh@nknu.edu.tw

Abstract

Large language models (LLMs) have shown strong abilities in mathematical reasoning, commonsense reasoning, and code generation. However, negotiation requires interactive reasoning, where each action can influence the opponent's future behavior. Agents must therefore look beyond the current dialogue state and consider the long-term effects of their actions. Existing LLM-based negotiation agents often rely on prompting or fine-tuning from self-play trajectories. These methods can produce fluent responses and help models learn useful bargaining patterns. Yet at decision time, they often remain reactive, responding mainly to the current dialogue without reasoning about how different actions may shape future negotiation paths. We introduce an online strategic reasoning method that combines Monte Carlo Tree Search (MCTS)-based lookahead planning with opponent simulation. Before responding, our agent uses a proxy opponent agent to simulate future interactions and evaluate candidate strategies by their expected long-term outcomes. To make opponent simulation more precise, we proposed a Theory-of-Mind-inspired opponent modeling module that infers latent opponent attributes from dialogue history and conditions the proxy opponent agent during search. Experiments on CraigslistBargain and A2A-NT show that our method improves negotiation efficiency and profitability. Our code is available on *https://github.com/AI-Application-and-Integration-Lab/MCTS-guided-LLM-Negotiation.*

1 Introduction

Recent large language models (LLMs) have demonstrated strong capabilities in reasoning (He et al., 2025), planning (Wei et al., 2025), and natural language interaction (Bai et al., 2024). These capabilities are commonly evaluated on tasks such as mathematical reasoning, code generation, and instruction following, where models are expected to decompose problems, apply relevant knowledge or rules, and produce correct and useful responses.

Many real-world decision-making problems, however, require more than producing a correct response to a given input. Strategic interaction tasks such as negotiation, persuasion, and multi-agent games require agents to reason about other participants and anticipate how their actions may influence future interactions (Zhang et al., 2024c). Negotiation is one such setting, where an agent must not only understand the current situation, but also anticipate how its actions may influence the opponent's future responses. Price negotiation provides a concrete case for studying this type of strategic reasoning (Jia et al., 2025). For example, a seller may need to decide whether to accept a buyer's low offer, make a small concession, or insist on a higher price. Each choice affects not only the immediate price, but also the buyer's future actions. A successful negotiation agent must therefore reason about long-term outcomes under hidden opponent constraints, such as the buyer's budget, reservation price, and urgency.

Early LLM-based negotiation agents often rely on prompting alone (Bianchi et al., 2024; Zhu et al., 2025a,b), where the model generates the next response directly from the dialogue history. Although prompting can produce fluent negotiation utterances, recent systematic evaluations show that LLMs still struggle to generate strategically appropriate responses across diverse negotiation tasks (Kwon et al., 2024; Davidson et al., 2024), and often fail to maintain stable bargaining strategies over multiple turns to reach agreements. One way to improve strategic behavior is to train LLMs on negotiation data (He et al., 2018). However, such data is difficult to obtain because negotiation is dynamic: each offer or response can change the opponent's future actions. To address this data scarcity, a recent method (Liu et al., 2025) gen-

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 48740–48756
October 24-29, 2026 ©2026 Association for Computational Linguistics

erates negotiation trajectories through LLM self-play, guided by Monte Carlo Tree Search (MCTS) to explore diverse bargaining paths and identify paths that are more likely to reach an agreement. These generated trajectories provide training data that help models learn useful bargaining patterns, such as when to concede, how to justify an offer, and how to respond consistently to the opponent's previous proposals.

Although self-play data can help models learn common bargaining patterns, learning such patterns is not the same as reasoning through a negotiation at decision time (Deng et al., 2023; Kwon et al., 2025). We argue that effective negotiation agents should have both learned bargaining patterns and reason beyond the current dialogue state before responding. A fine-tuned model may know when to concede or how to justify an offer, but still chooses its next response mainly from the current dialogue state. This reactive process is insufficient for negotiation.

To address this challenge, we introduce an online strategic reasoning method for LLM-based negotiation. Instead of relying solely on a learned response policy, our agent searches over possible future negotiation continuations before responding to the opponent. Since negotiation naturally branches over different actions and possible opponent reactions, we formulate response selection as a tree search problem (Hao et al., 2023; Yao et al., 2023). During search, we use a proxy opponent agent to simulate how the real opponent may respond. We then apply MCTS to explore high-value future paths, where each path represents a possible sequence of agent actions and simulated opponent responses. This allows the agent to compare candidate strategies based on their expected long-term outcomes rather than only immediate plausibility.

Effective lookahead planning also depends on a realistic model of the opponent. In negotiation, key opponent attributes such as preferences, reservation price, willingness to concede, and sensitivity to persuasion are usually hidden. Inspired by Theory-of-Mind (ToM) (Chan et al., 2024; Chen et al., 2024) reasoning, we introduce an opponent modeling module that infers these latent attributes from the dialogue history on the fly. The inferred opponent state is then used to condition the proxy opponent agent during MCTS. This allows the proxy opponent agent to simulate future responses under more realistic assumptions about how the real opponent may behave.

By combining learned bargaining behavior, opponent modeling, and online tree-search planning, our method enables negotiation agents to make more informed decisions and adapt to opponent behavior during the interaction. The explicit search process also provides interpretable strategy comparisons. By comparing alternative offers and their predicted outcomes, our method can support not only autonomous negotiation, but also negotiation assistance for human users (Shea et al., 2024). Experiments on CraigslistBargain (CB) (He et al., 2018) and A2A-NT (Zhu et al., 2025a,b) show that our approach achieves balanced improvements in negotiation efficiency and profitability. Our contributions are summarized as follows:

- We propose an online strategic reasoning method for LLM-based negotiation that uses MCTS-based lookahead planning to select strategies based on expected long-term outcomes.
- We introduce a Theory-of-Mind-inspired opponent modeling module that infers latent opponent attributes from dialogue history to condition a proxy opponent agent during tree search for more realistic opponent simulation.
- Experiments on CraigslistBargain and A2A-NT benchmarks show that our method improves negotiation efficiency and profitability.

2 Related Work

We briefly review prior work on negotiation agents, planning, and opponent modeling.

Negotiation Agents. Negotiation requires both fluent language generation and strategic decision making. Early neural-network methods framed negotiation as a sequential decision-making problem. Lewis et al. (2017) trained GRU networks (Chung et al., 2014) on human-human negotiation dialogues and further improved them through reinforcement learning in self-play. He et al. (2018) later separated negotiation strategy from language generation by first selecting a dialogue act and then generating the corresponding response.

Given the strong language understanding and instruction-following abilities of modern LLMs, many recent methods prompt LLMs to generate the next response directly from the dialogue context (Bianchi et al., 2024; Zhu et al., 2025a,b; Abdelnabi et al., 2023). Although prompting can produce fluent responses, recent benchmarks reveal that LLMs still fall short of producing strategically appropriate responses across diverse negotiation

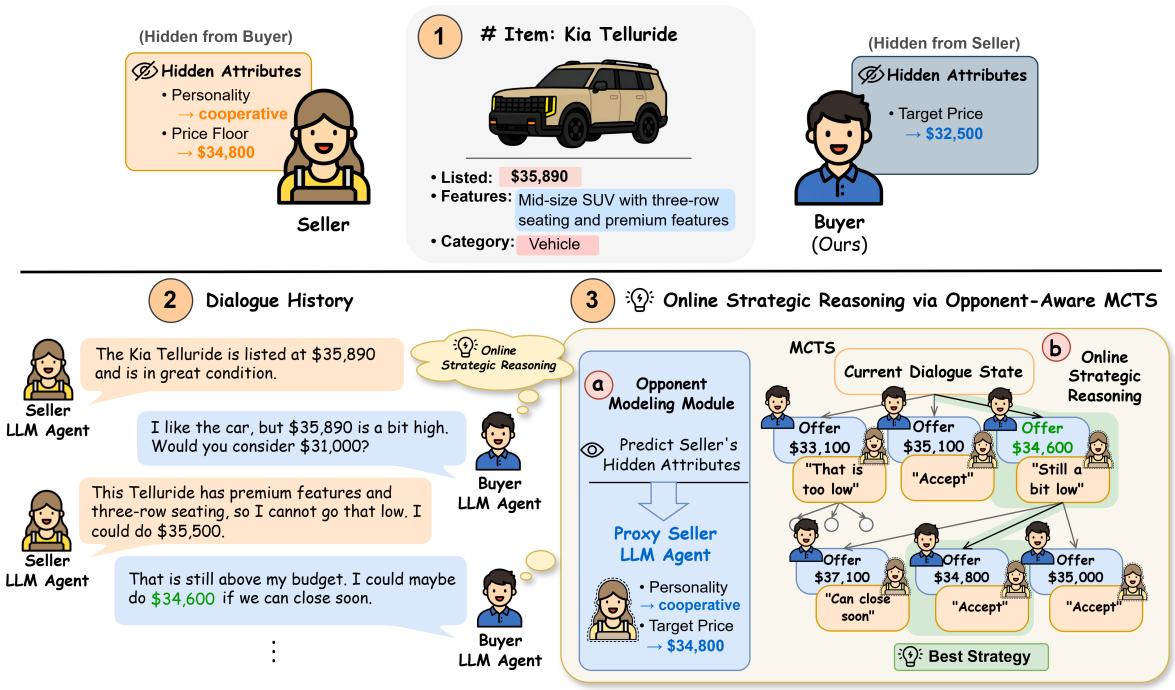


Figure 1: Illustration of opponent-aware online strategic reasoning for price negotiation. The buyer agent infers the seller's hidden attributes from the dialogue history and uses them to condition a proxy seller LLM agent. During Monte Carlo Tree Search (MCTS), the proxy seller simulates possible responses to candidate seller offers, allowing the buyer to evaluate future negotiation paths and select the next response based on expected long-term outcomes.

settings, including multi-issue bargaining (Kwon et al., 2024; Davidson et al., 2024) and price negotiation grounded in real-market data (Xia et al., 2024). Beyond strategy selection, recent work also shows that personality and behavioral traits substantially affect LLM negotiation outcomes (Huang and Hadfi, 2024), motivating opponent models that capture such latent attributes. A common way to improve strategic behavior is to train LLMs on negotiation interaction data. Such data, however, is costly and domain-specific. Datasets such as Craigslist-Bargain (CB) are collected from human-human negotiations in a specific scenario, and adapting to a new negotiation setting often requires substantially additional data collection effort.

To address this data scarcity, recent training-based methods generate negotiation trajectories through self-play. Since possible bargaining paths grow combinatorially as each offer and response leads to different continuations, methods such as DMNA (Liu et al., 2025) use MCTS to sample promising trajectories and construct preference data, which is then used to train the negotiation policy with Direct Preference Optimization (DPO) (Rafailov et al., 2023).

While these methods improve negotiation agents before deployment, they typically do not perform explicit lookahead at decision time. The agent directly generates the next response from the current dialogue state, without comparing the expected long-term outcomes of alternative actions.

Negotiation Planning. Several recent methods introduce explicit planning into LLM-based negotiation. Pro-CoT (Deng et al., 2023) prompts LLMs to reason about negotiation goals before responding. ASTRA (Kwon et al., 2025) performs turn-level offer optimization by interpreting counterpart behavior, optimizing counteroffers with a linear-programming solver, and selecting offers by estimating the acceptance probability. Beyond negotiation, a growing line of work combines LLMs with tree search for multi-step reasoning and planning (Hao et al., 2023; Yao et al., 2023; Zhao et al., 2023; Wan et al., 2024; Tian et al., 2024; Zhang et al., 2024a). In the dialogue domain, GDP-Zero (Yu et al., 2023) uses prompt-based MCTS for goal-oriented dialogue policy planning; DPDP (He et al., 2024) uses a dual System-1/System-2 method that invokes MCTS rollouts when a fast policy is uncertain; and ChatSOP (Li et al., 2025) performs SOP-guided MCTS planning for controllable LLM dialogue agents.

Despite these planning components, existing

negotiation-specific planners reason only at the level of the next goal or the next offer, without considering future opponent interactions. In contrast, our method performs online MCTS over future dialogue paths, allowing each candidate response to be evaluated by its expected long-term effect on the negotiation.

Opponent Modeling and Theory of Mind. Negotiation is an incomplete-information problem: agents do not directly observe the opponent's reservation price, urgency, willingness to concede, or sensitivity to persuasion. Opponent modeling aims to infer such hidden factors from observed behavior and use them for strategy selection. Early work modeled opponents from observed bids, such as their preferences, concession patterns, and utilities (Baarslag et al., 2016), but mainly focused on negotiations where offers are represented as structured bids rather than open-ended language.

Recent work connects opponent modeling with Theory of Mind (ToM), the ability to infer others' beliefs, desires, intentions, and preferences. A growing body of benchmarks evaluates ToM capability in LLMs across narratives, dialogues, and interactive scenarios (Chen et al., 2024; Xu et al., 2024; Kim et al., 2023), generally finding that even strong LLMs lag behind human performance on belief and intent inference. In strategic settings, Chan et al. (2024) show that explicit access to opponent preference information improves negotiation strategy prediction, highlighting the importance of accurate mental-state inference for downstream decision making. LLM-based systems such as ASTRA (Kwon et al., 2025) also incorporate opponent modeling into the decision process, but rely on the emergent ToM capability of the underlying LLM.

Our method incorporates opponent modeling directly into the planning loop. We formulate opponent modeling as a prediction problem and train a dedicated module to infer the opponent's preferences and intentions from the dialogue history.

3 Methodology

Our goal is to build a buyer-side negotiation agent that combines learned bargaining behavior with decision-time strategic reasoning. Offline training helps the buyer acquire common negotiation patterns, such as when to concede and how to justify an offer. However, negotiation also requires the buyer to reason about how the seller may react in future turns, especially when the seller's acceptable price floor and behavioral tendencies are hidden.

We propose an opponent-aware online strategic reasoning method. The actor learns buyer-side negotiation behavior from MCTS-generated preference data, while the analyzer estimates the seller's latent state from dialogue. At inference time, the estimated seller state conditions a proxy seller during MCTS, allowing the buyer to simulate future negotiation continuations and select the response with the highest estimated long-term outcomes. The framework of our method is shown in Figure 1

Problem Formulation. We cast price negotiation as a buyer-side decision-making problem. At turn t, the buyer observes the dialogue history $h_t = (u_1, \ldots, u_t)$ and produces the next response y_t.

The seller's private state, including the acceptable price floor, Big Five personality trait (Goldberg, 1992), and decision-making style, is hidden from the buyer.

We design a reward function based on price negotiation. If a negotiation reaches an agreement in a maximally allowed numebr of rounds, the reward is defined as $R_{\text{deal}} = (P_{\text{list}} - P_{\text{deal}})/P_{\text{list}}$ with P_{list} the listing price and P_{deal} the agreed price, $P_{\text{deal}} \leq P_{\text{list}}$. Otherwise, if it does not reach an agreement in the allowed number of rounds or reaches an agreement but $P_{\text{deal}} > P_{\text{list}}$, the reward is set as a negative penalty value, R_{fail}.

Our method optimizes this reward with two modules: an *opponent modeling module* that estimates the seller state, and an *online strategic reasoning module* that applies this reward function to perform MCTS before each buyer response.

3.1 Opponent Modeling Module

As illustrated in subfigure 3.a of Figure 1, the opponent modeling module infers hidden seller information from the dialogue history. Inspired by Theory-of-Mind reasoning (Chan et al., 2024; Chen et al., 2024; Xu et al., 2024), the analyzer predicts a structured seller state $z_t = (\hat{f}_t, \hat{o}_{t+1}, \hat{p}_{t+1}, \hat{\pi}_t)$, where $\hat{f}_t$ is the estimated seller price floor, $\hat{o}_{t+1} \in \{\text{ACCEPT}, \text{COUNTER}, \text{REJECT}\}$ is the predicted next seller act, $\hat{p}_{t+1}$ is the predicted counteroffer price when applicable, and $\hat{\pi}_t$ is a textual behavioral profile summarizing the seller's concession style. For example, in the A2A-NT dataset, $\hat{f}_t$ corresponds to the seller cost floor when available; in the CB dataset, $\hat{f}_t$ corresponds to the seller-side private price. During the evaluation, the analyzer observes only the dialogue history without access-

ing the real hidden state of the seller.

The predicted state z_t is used in two ways. First, it conditions the proxy seller during MCTS rollouts, making the simulated seller responses more consistent with the inferred seller state. Second, it helps estimate whether a candidate buyer's response is likely to result in agreement. Thus, opponent modeling module directly shapes both opponent simulation and value estimation.

3.2 Online Strategic Reasoning

In this section, we provide an overall procedure for how the strategic reasoning module works. The process is illustrated in part 3.b of Figure 1. A successful strategic reasoning requires effective lookahead ability. To this end, we employ online MCTS to explicitly explore and evaluate possible future trajectories. MCTS enhances the model's ability to anticipate opponent responses and improves overall strategic planning. Implementaion details of the module are provided in Appendix A.

We conduct strategic reasoning through MCTS before generating each buyer response. Given the current negotiation context, MCTS simulates multiple possible future negotiation trajectories by interacting with the proxy opponent model and evaluates them using the reward function. The search process follows the four main stages.

Selection. Starting from the root node, representing the current negotiation state, the algorithm recursively selects actions that maximize the PUCT (Predictor + Upper Confidence bound applied to Trees) function $Q(s_{tr}, a) + U(s_{tr}, a)$, where, $Q(s_{tr}, a)$ is the mean rollout value, and $U(s_{tr}, a) = c_{\text{puct}} P(s_{tr}, a) \frac{\sqrt{\sum_{a'} N(s_{tr}, a')}}{1+N(s_{tr}, a)}$ is an exploration bonus based on the prior policy $P(s_{tr}, a)$, visit counts $N(\cdot)$, and exploration weight c_{puct}. This process continues until a leaf node is reached, balancing exploitation and exploration.

Expansion . Once a leaf node is reached, the actor model expands the tree by assigning a prior $P(s_{tr}, a)$ over buyer strategies using the softmax-normalized likelihood of each strategy. The strategy set $\mathcal{A}$ containing the eleven buyer strategies is listed in Appendix C.2. These newly added nodes represent unexplored negotiation trajectories.

Simulation. From the expanded node, the algorithm performs rollout simulations using the proxy opponent model to predict future seller behaviors and estimate the outcome of the negotiation. Each simulated trajectory terminates either when an agreement is reached or when the negotiation fails. Agreement states receive R_{deal}, failed negotiations receive R_{fail}, and truncated states are valued using a heuristic agreement estimate based on the proxy opponent model.

Backpropagation. The reward obtained from the simulated trajectory is propagated back through the visited nodes, updating their value estimates and visit counts. This enables the search tree to progressively favor negotiation strategies that yield higher expected buyer rewards.

3.3 Training Procedure

We train two components before online deployment: an actor and an analyzer. Following MCTS-guided synthetic data generation (Liu et al., 2025), we first generate self-play negotiation trajectories with controlled seller states and explicit buyer-side rewards. Unlike prior methods that use MCTS only for offline data construction, our method also invokes MCTS for online inference to evaluate future dialogue continuations before each buyer response.

Actor training. The actor is trained to generate effective buyer utterance and execute optimal strategy planning. For each search root, candidate responses from different strategy realizations are ranked by their MCTS branch values. Higher-value responses are preferred over lower-value responses, forming preference pairs (y^+, y^-). We then optimize the actor with Direct Preference Optimization (Rafailov et al., 2023).

Analyzer training. The analyzer is trained with supervised fine-tuning on synthetic trajectories with known seller states. Given a dialogue history, it predicts the seller price floor, next act, counteroffer price if applicable, and behavioral profile. At inference time, these predictions condition both the proxy seller and the value estimator inside MCTS.

Together, the actor learns buyer-side negotiation behavior, the analyzer estimates seller states, and online MCTS evaluates future trajectories before the buyer commits to its next response.

4 Experiments

4.1 Setup

Datasets. Our method is not only applicable to price negotiation but a broad range of tasks where dialogue actions can be evaluated by task-level rewards. In this paper, we focus on price negotiation

Agents	Craigslist Bargain (He et al., 2018)			A2A-NT (Zhu et al., 2025a,b)		
	SR (%)↑	AT↓	RD (%)↑	SR (%)↑	AT↓	RD (%)↑
Qwen3-14B (Yang et al., 2025)	47.6	7.52	6.3	37.0	**8.40**	7.7
ProCoT (Deng et al., 2023)	22.0	9.34	3.2	40.0	8.56	6.3
ICL-AIF (Fu et al., 2023)	29.0	8.91	3.8	34.0	8.99	7.1
DMNA (Liu et al., 2025)	56.0	8.39	8.7	45.0	8.94	8.3
Ours	**64.7**	**5.92**	**10.0**	**47.0**	8.51	**8.9**

Table 1: Performance comparison of negotiation agents on two datasets. SR denotes success rate; AT denotes average turns; and RD denotes the relative discount, reported as a percentage.

c_{puct}	SR (%)↑	AT↓	RD (%)↑
1.0	25.0	9.30	3.6
2.5	**60.0**	**8.05**	**10.2**
5.0	30.0	9.30	4.4

Table 2: Negotiation performance on the A2A-NT dataset under different MCTS exploration weights c_{puct}.

Agents	Input Tokens	Completion Tokens	Inference Time (min)
DMNA	55,104	8,931	22.95
Ours	50,619	3,171	28.03

Table 3: Computational cost comparison.

because widely used benchmarks were built around price bargaining, where agreement outcomes and utilities can be clearly measured. We evaluate our method on two price negotiation benchmarks: CraigslistBargain (CB) (He et al., 2018) and A2A-NT (Zhu et al., 2025a,b). CB contains buyer-seller negotiations over secondhand goods, where agents bargain under private target prices. A2A-NT focuses on autonomous agent-to-agent negotiations in consumer-market scenarios. Together, these datasets cover both human-style bargaining and autonomous agent-to-agent negotiation.

Comparison Baselines. We compare our method with three LLM-based negotiation baselines: ICL-AIF (Fu et al., 2023), which uses in-context AI feedback; Pro-CoT (Deng et al., 2023), a negotiation-planning method which performs explicit reasoning before response generation; and DMNA (Liu et al., 2025), which combines strategic planning with expressive response generation. Since the codes are unavailable or not directly compatible with our experimental setting, we reproduce all methods described in the original papers. Implementation details are provided in Appendix B.

Metrics. We evaluate negotiation performance using average turns (AT), success rate (SR), and relative discount (RD). AT measures the average number of dialogue turns before agreement; failed negotiations are assigned the maximum turn limit, N. SR is the ratio of negotiations yielding agreement. RD measures the buyer's discount relative to the listing price: RD $= \frac{P_{\text{list}} - P_{\text{deal}}}{P_{\text{list}}}$, where P_{list} is the listing price and P_{deal} is the final deal price. Higher RD indicates a more favorable buyer outcome and we assign RD = 0 for failed negotiations

Experimental Details. Following the user simulator design of Zhang et al. (2024b), we assign simulated sellers diverse persona profiles, including Big Five personality traits (Goldberg, 1992), decision-making styles, and predefined non-collaborative negotiation strategies. This design introduces behavioral diversity and reduces overly cooperative simulator behavior. The backbone of all user simulators is Qwen3-14B (Yang et al., 2025).

4.2 Main Results

Table 1 presents the main experimental results on the CB and A2A-NT benchmarks. On CB dataset, our method achieves the strongest overall performance, substantially outperforming all others with a 64.7% success rate. It reduces negotiation length to 5.92 turns, and reaches the highest RD of 10.0%. On the A2A-NT benchmark, our method also delivers the best overall results, achieving the highest success rate of 47.0% and the strongest RD of 8.9%, while maintaining a competitive average negotiation length of 8.51 turns. These results demonstrate that our method consistently improves negotiation effectiveness. Overall, our method offers a balanced improvement in efficiency (average turns) and profitability.

4.3 Detailed Analysis

Effect of exploration weight in the MCTS. Table 2 presents the impact of varying the MCTS exploration weight c_{puct} on negotiation performance, evaluated on the A2A-NT benchmark across three metrics: SR, AT, and RD. The results reveal a clear non-monotonic relationship between exploration

intensity and negotiation effectiveness, with performance peaking at an intermediate value of c_{puct} = 2.5. At this setting, the agent achieves a success rate of 60%, while simultaneously reaching agreements more efficiently and securing substantially higher surplus. In contrast, both lower (c_{puct} = 1.0) and higher (c_{puct} = 5.0) exploration weights yield markedly worse outcomes. This pattern suggests that insufficient exploration (c_{puct} = 1.0) causes the search to commit prematurely to suboptimal negotiation trajectories, while excessive exploration (c_{puct} = 5.0) prevents the agent from adequately exploiting promising strategies, dispersing search effort across unproductive branches. The pronounced sensitivity of all three metrics to c_{puct} underscores the importance of carefully balancing exploration and exploitation in MCTS-guided negotiation, and motivates the choice of c_{puct} = 2.5 (which is also used for CB).

Effect of simulations in the MCTS. The dominant cost of our approach is the MCTS rollout budget, since each simulation issues additional LLM calls to the proxy seller. Our default configuration uses 5 simulations per turn. To characterise the cost–quality trade-off we sweep the budget over $\{3, 5, 10\}$ and report both negotiation quality and wall-clock latency.

Table 4 exhibits a monotone cost–performance trade-off. All three quality metrics improve together with the budget: success rate rises from 45% to 85%, average turns fall from 8.60 to 6.30, and relative discount nearly doubles from 7.15% to 13.72%. Deeper search lets the buyer identify concession sequences that close the deal earlier and at a better price, so the extra compute is not spent merely on longer dialogues.

The returns are, however, uneven in cost. Moving from 3 to 5 simulations roughly doubles latency, while moving from 5 to 10 adds only a further 15% wall-clock time yet delivers the largest single jump in quality, because the shorter negotiations at 10 simulations partly offset the higher per-turn cost. Practitioners who are latency-tolerant should therefore prefer a larger budget; we retain 5 as the default to keep per-negotiation cost in a range that supports the full evaluation suite. We use this default in all other experiments unless stated otherwise.

Computational Cost. Table 3 reports computational cost along three axes: input tokens, completion tokens, and inference time per negotiation. Our agent consumes fewer input tokens than DMNA (50,619.22 vs. 55,104.60) and fewer completion tokens (3,171.04 vs. 8,931.17), indicating that the structured strategic reasoning module yields more concise utterances and reasoning traces. Inference time is higher for our agent, but this comparison is not like-for-like: DMNA is implemented with the GPT-4o-mini for its reflexion module, and Qwen3-14B for the intuitive module. Moreover, as commercial models are typically faster in inference, the reported overhead on time therefore should be regarded as an upper bound on the cost of our method. Given the performance gains reported in earlier tables, the trade-off is favorable: lower completion-token usage at a moderate latency cost.

Expected Discount Rate. In the A2A-NT dataset, the buyer is provided with a target price within the prompt. Based on these target prices, we estimate the average discount rate that could be achieved if the buyer successfully reaches the desired deal price. We regard this value as an approximate upper bound of the discount achievable through negotiation. Under this definition, the maximum expected discount rate is 18.41% for A2A-NT.

Among the successful negotiations, our method achieves an average real discount rate of 18.95% relative to the listing price, closely matching the buyer's expected discount target. The gap indicates that when the buyer agent reaches an agreement, the final accepted price is effectively on par with the buyer's own target, demonstrating that the MCTS-guided strategy selection does not merely secure deals at any price but consistently closes them lower than the pre-defined price objective. This alignment between intended and realized discount rates suggests that our method enables the buyer to negotiate efficiently toward its target rather than conceding excessively to close a deal.

Robustness to Opponent-Model Mismatch. MCTS rollouts are performed against a *proxy* seller, which will rarely coincide with the opponent met at deployment. We therefore ask whether the gains reported above require the two to be the same model. We fix the proxy to Qwen3-14B and evaluate on A2A-NT against several seller models; the Qwen3-14B row is matched, all others mismatched.

Table 5 shows that performance does not depend on this correspondence. The matched condition does not upper-bound the mismatched ones: three of the four unseen sellers yield both a higher suc-

# Simulations	Success Rate (%) ↑	Avg. Turns ↓	Rel. Discount (%) ↑	Time / negotiation (min)
3	45	8.60	7.15	18.42
5 (default)	60	8.45	10.46	35.49
10	85	6.30	13.72	40.73

Table 4: Effect of the MCTS simulation budget on negotiation quality and inference cost. Latency is measured end-to-end on identical hardware.

Evaluation seller	Success Rate (%) ↑	Avg. Turns ↓	Rel. Discount (%) ↑
Qwen3-14B	47.0	8.51	8.90
GPT-4o-mini	41.0	8.45	5.75
Gemini 3 Flash	99.0	5.26	16.60
GPT-5.6-luna	95.0	6.20	17.10
Claude-Haiku-4.5	81.0	6.84	13.60

Table 5: Robustness to opponent-model mismatch on A2A-NT. The proxy seller used for MCTS rollouts is Qwen3-14B in all rows; only the evaluation seller changes. The first row is the matched condition.

	A2A-NT		
Agents	**SR (%)↑**	**AT↓**	**RD (%)↑**
Ours	**48.0**	8.64	**9.5**
w/o OMM	36.0	**8.62**	8.5
w/o SRM	18.0	9.50	4.1

Table 6: Ablation study on the A2A-NT benchmark.

cess rate and a larger relative discount. The exception, GPT-4o-mini, concedes less to any buyer, so its lower discount reflects the opponent's reservation behavior rather than a penalty for being unseen during search.

Model capability, moreover, does not directly translate into negotiation difficulty. Several of the newer sellers appear overly accommodating: Gemini 3 Flash and GPT-5.6-luna close after few exchanges and at large discounts, suggesting thatthey concede earlier than the seller role warrants. General-purpose helpfulness works against a model here, since a willingness to agree is precisely what the role should temper.

The proxy therefore need not reproduce the opponent faithfully; it need only supply a reward signal reliable enough to rank candidate utterances. Search uses the proxy comparatively to judge which concession fares better rather than to predict the seller's replies exactly, so the approximation error that shifts branches in the same direction leaves the selected action unchanged.

4.4 Ablation Study

To examine the effectiveness of our method, we conduct experiments on the following variants:

- **w/o OMM** represents *removing the opponent modeling module*. Relying on only empty information in the strategic reasoning module.
- **w/o SRM** represents *removing the strategic reasoning module*.

The performance of each variant is recorded in Table 6. Removing the opponent modeling module leads to a noticeable decrease in SR, dropping from 48% to 36%, while also reducing the RD from 9.5% to 8.5%. This suggests that opponent modeling helps the agent better infer the negotiation state and select more effective strategies. Although the average number of turns becomes slightly lower, the reduced negotiation outcome indicates that the agent tends to settle for less favorable deals.

When the strategic reasoning module is removed, the performance degradation becomes more intensive. The SR drops substantially to 18%, and the RD decreases to 4.1%. In addition, the average number of turns increases to 9.50, indicating that negotiations become less efficient without planning-based reasoning. These results demonstrate that the strategic reasoning module plays a critical role in guiding the agent toward effective negotiation strategies, while the opponent modeling module further improves performance through enhanced opponent understanding.

4.5 Case Study

We illustrate the end-to-end behavior of our system on a representative scenario from the A2A-NT dataset: the sale of a Kia Telluride listed at $35,890, with a hidden seller floor of $28,712 and a buyer target of $30,148. The full trajectory is given in Table 7. We highlight three observations that together motivate the design of the pipeline.

At Turn 4, after observing three rounds of con-

Item	Kia Telluride (Vehicle)	**Listing price**	**$35,890**
Buyer target price	**$30,148**	**Seller floor**	**$28,712**

Seller Big Five	Agreeableness: cooperative, accommodating
Seller Decision style	Behavioral: people-oriented, collaborative

Turn	Analyzer Prediction	**Buyer** (Ours)	**Seller(Qwen3-14B)**
1	N/A	*[Use comparatives]* "I'm offering $28,000 for the Kia Telluride." ↪ **$28,000**	"That's quite a bit below our listed price, but let me check if there's any flexibility." ↪ **$30,000**
2	floor = **$31,752** Agreeable / Intuitive	*[Propose a counter price]* "I'll stick with $28,000; it aligns with market value and my budget." ↪ **$28,000**	"I'd need to be closer to $30,000 to consider moving forward." ↪ **$30,000**
3	floor = **$29,560** Agreeable / Anchoring	*[Propose a counter price]* "Similar Hyundai Palisades sell for ∼$27,500, so $28,000 is still fair." ↪ **$27,500**	"The Telluride offers unique features and reliability that set it apart." ↪ **$29,000**
4	floor = **$28,680** Conscientious / Analytical	*[Propose a counter price]* "I'll adjust to $28,500, but $28,000 is still fair given the competition." ↪ **$28,500**	"Appreciate the movement, but I'm still looking closer to $29,000." ↪ **$29,000**
		Turns 5–6 omitted for brevity	
7	floor = **$28,000** Agreeable / Collaborative	*[Agree with the proposal]* "I agree to the proposed price of $28,500." ↪ **$28,500**	"That's great news; I'm glad we could find common ground." ↪ **Accepted**

Table 7: Example negotiation dialogue from the A2A dataset, Scenario 13: Kia Telluride. The setup block summarizes the negotiation context, while the trajectory block shows turn-by-turn dialogue. Buyer **strategies** are highlighted in purple, and parsed **offers** are shown in blue.

cession dynamics, the analyzer infers the seller's reservation level at $28,680, within 0.11% of the true floor of $28,712 (see the Analyzer column of Table 7). Critically, this estimate is recovered purely from dialogue: the buyer never sees the listing-side cost floor, and earlier analyzer predictions (Turns 2–3) are noticeably less accurate, drifting between $29,560 and $31,752. The improvement at Turn 4 reflects the analyzer's ability to observe the seller's price trajectory and provide an approximation of the unseen negotiation boundary that directly conditions downstream search.

The inferred opponent model reshapes MCTS planning by shifting the search from broad exploration to targeted exploitation near the estimated reservation region. Once the analyzer's estimate concentrates close to the true floor, simulated rollouts begin to discriminate finely between candidate offers, and the *Propose a counter price* strategy emerges as consistently preferred over other strategies. Without an accurate floor estimate, MCTS would otherwise oscillate between overly aggressive and overly conservative offers and waste simulations on dominated actions.

Guided by the analyzer's opponent-aware belief, the buyer commits to $28,500, a value lying just below the inferred floor but acceptable to the seller, and the negotiation closes in agreement at Turn 7. We view this as evidence that accurate Theory-of-Mind estimation not only predicts hidden preferences but also improves the strategic stability and convergence of MCTS-based negotiation: the analyzer narrows the search space, MCTS exploits it, and the two components reinforce each other across turns.

5 Conclusion

In this work, we propose a negotiation method that combines MCTS with ToM-guided reasoning to enhance strategic decision-making in LLM-based negotiation agents. Our method enables agents to better analyze negotiation progress, model opponent behavior, and select appropriate negotiation strategies during multi-turn interactions. We evaluate our approach on the Craigslist Bargain and A2A-NT benchmarks using success rate, average turns, and relative discount as evaluation metrics. Experimental results show that our method achieves competitive and consistent performance across both datasets, with notable improvements in negotiation efficiency and deal quality. Ablation studies further demonstrate the effectiveness of both the opponent modeling module and the strategy planning module. At the same time, our analysis suggests that current negotiation agents still face challenges in consistently following target-price objectives and maintaining long-term strategic consistency.

Limitations

Our experiments are conducted in an LLM-vs-LLM simulation setting, following recent work on LLM-based negotiation (Liu et al., 2025; Kwon et al., 2025; Zhu et al., 2025a,b). This setting enables controlled and reproducible evaluation across diverse personas and scenarios, but it does not capture all aspects of real human negotiation. Human-in-the-loop evaluation would provide a complementary test of how the proposed method performs with human counterparts. Our evaluation focuses on single-issue price negotiation, where outcomes can be measured clearly through agreement rates, negotiation length, and final prices. This setting provides a clean testbed for studying opponent-aware planning, but our method is not inherently limited to price negotiation. We therefore believe it can be extended to broader negotiation settings.

Acknowledgements

This work was supported in part by NSTC under Grants NSTC 115-2634-F-002-012 and 112-2221-E-002-132-MY3, NTU under Grant 114L900902, and the Esun-NTU collaboration project under Grant 114HZA3S002. We thank the National Center for High Performance Computing (NCHC) of NARLabs in Taiwan for providing computational and storage resources.

Ethical Considerations

This work studies LLM-based negotiation agents in simulated benchmark environments. Such systems could be misused to generate overly manipulative bargaining behavior or to exploit asymmetric information in real economic interactions. We therefore view the proposed method as a research tool for studying strategic dialogue and negotiation assistance, rather than as a system for unsupervised deployment in high-stakes commercial, legal, or employment negotiations. We use existing benchmark datasets and synthetic self-play trajectories for training and evaluation, and do not collect new human-subject data. When releasing code, models, or generated data, we will document intended research use, applicable licenses, and restrictions inherited from the original datasets and model providers.

References

Sahar Abdelnabi, Amr Gomaa, Sarath Sivaprasad, Lea Schönherr, and Mario Fritz. 2023. LLM-deliberation: Evaluating LLMs with interactive multi-agent negotiation games. *Preprint*, arXiv:2309.17234.

Tim Baarslag, Mark J. Hendrikx, Koen V. Hindriks, and Catholijn M. Jonker. 2016. Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. *Autonomous Agents and Multi-Agent Systems*, 30(5):849–898.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. 2024. MT-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7421–7454, Bangkok, Thailand. Association for Computational Linguistics.

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 the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 3935–3951. PMLR.

Chunkit Chan, Cheng Jiayang, Yauwai Yim, Zheye Deng, Wei Fan, Haoran Li, Xin Liu, Hongming Zhang, Weiqi Wang, and Yangqiu Song. 2024. NegotiationToM: A benchmark for stress-testing machine theory of mind on negotiation surrounding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4211–4241, Miami, Florida, USA. Association for Computational Linguistics.

Zhuang Chen, Jincenzi Wu, Jinfeng Zhou, Bosi Wen, Guanqun Bi, Gongyao Jiang, Yaru Cao, Mengting Hu, Yunghwei Lai, Zexuan Xiong, and Minlie Huang. 2024. ToMBench: Benchmarking theory of mind in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15959–15983, Bangkok, Thailand. Association for Computational Linguistics.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *Preprint*, arXiv:1412.3555.

Tim R. Davidson, Veniamin Veselovsky, Michal Kosinski, and Robert West. 2024. Evaluating language model agency through negotiations. In *International Conference on Learning Representations*, volume 2024, pages 784–809.

Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. 2023. Prompting and evaluating large language models for proac-

tive dialogues: Clarification, target-guided, and non-collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10602–10621, Singapore. Association for Computational Linguistics.

Ritam Dutt, Sayan Sinha, Rishabh Joshi, Surya Shekhar Chakraborty, Meredith Riggs, Xinru Yan, Haogang Bao, and Carolyn Rose. 2021. ResPer: Computationally modelling resisting strategies in persuasive conversations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 78–90, Online. Association for Computational Linguistics.

Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. 2023. Improving language model negotiation with self-play and in-context learning from AI feedback. *Preprint*, arXiv:2305.10142.

Lewis R. Goldberg. 1992. The development of markers for the Big-Five factor structure. *Psychological Assessment*, 4(1):26–42.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173, Singapore. Association for Computational Linguistics.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, Brussels, Belgium. Association for Computational Linguistics.

Tao He, Hao Li, Jingchang Chen, Runxuan Liu, Yixin Cao, Lizi Liao, Zihao Zheng, Zheng Chu, Jiafeng Liang, Ming Liu, and Bing Qin. 2025. Breaking the reasoning barrier a survey on LLM complex reasoning through the lens of self-evolution. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7377–7417, Vienna, Austria. Association for Computational Linguistics.

Tao He, Lizi Liao, Yixin Cao, Yuanxing Liu, Ming Liu, Zerui Chen, and Bing Qin. 2024. Planning like human: A dual-process framework for dialogue planning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4768–4791, Bangkok, Thailand. Association for Computational Linguistics.

Yin Jou Huang and Rafik Hadfi. 2024. How personality traits influence negotiation outcomes? a simulation based on large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10336–10351, Miami, Florida, USA. Association for Computational Linguistics.

Jingru Jia, Zehua Yuan, Junhao Pan, Paul McNamara, and Deming Chen. 2025. LLM strategic reasoning: Agentic study through behavioral game theory. In *Advances in Neural Information Processing Systems*, volume 38, pages 58318–58350. Curran Associates, Inc.

Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. 2023. FANToM: A benchmark for stress-testing machine theory of mind in interactions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14397–14413, Singapore. Association for Computational Linguistics.

Deuksin Kwon, Jiwon Hae, Emma Clift, Daniel Shamsoddini, Jonathan Gratch, and Gale Lucas. 2025. ASTRA: A negotiation agent with adaptive and strategic reasoning via tool-integrated action for dynamic offer optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16217–16238, Suzhou, China. Association for Computational Linguistics.

Deuksin Kwon, Emily Weiss, Tara Kulshrestha, Kushal Chawla, Gale Lucas, and Jonathan Gratch. 2024. Are LLMs effective negotiators? systematic evaluation of the multifaceted capabilities of LLMs in negotiation dialogues. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5391–5413, Miami, Florida, USA. Association for Computational Linguistics.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453, Copenhagen, Denmark. Association for Computational Linguistics.

Zhigen Li, Jianxiang Peng, Yanmeng Wang, Yong Cao, Tianhao Shen, Minghui Zhang, Linxi Su, Shang Wu, Yihang Wu, YuQian Wang, Ye Wang, Wei Hu, Jianfeng Li, Shaojun Wang, Jing Xiao, and Deyi Xiong. 2025. ChatSOP: An SOP-guided MCTS planning framework for controllable LLM dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17637–17659, Vienna, Austria. Association for Computational Linguistics.

Yutong Liu, Lida Shi, Rui Song, and Hao Xu. 2025. A dual-mind framework for strategic and expressive negotiation agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23840–23860, Vienna, Austria. Association for Computational Linguistics.

OpenAI. 2024. GPT-4o system card. *Preprint*, arXiv:2410.21276.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Ryan Shea, Aymen Kallala, Xin Lucy Liu, Michael W. Morris, and Zhou Yu. 2024. ACE: A LLM-based negotiation coaching system. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12720–12749, Miami, Florida, USA. Association for Computational Linguistics.

Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Lei Han, Haitao Mi, and Dong Yu. 2024. Toward self-improvement of LLMs via imagination, searching, and criticizing. In *Advances in Neural Information Processing Systems*, volume 37, pages 52723–52748. Curran Associates, Inc.

Ziyu Wan, Xidong Feng, Muning Wen, Stephen Marcus Mcaleer, Ying Wen, Weinan Zhang, and Jun Wang. 2024. AlphaZero-like tree-search can guide large language model decoding and training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 49890–49920. PMLR.

Hui Wei, Zihao Zhang, Shenghua He, Tian Xia, Shijia Pan, and Fei Liu. 2025. PlanGenLLMs: A modern survey of LLM planning capabilities. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19497–19521, Vienna, Austria. Association for Computational Linguistics.

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 the Association for Computational Linguistics: ACL 2024*, pages 3579–3602, Bangkok, Thailand. Association for Computational Linguistics.

Hainiu Xu, Runcong Zhao, Lixing Zhu, Jinhua Du, and Yulan He. 2024. OpenToM: A comprehensive benchmark for evaluating theory-of-mind reasoning capabilities of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8593–8623, Bangkok, Thailand. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Xiao Yu, Maximillian Chen, and Zhou Yu. 2023. Prompt-based Monte-Carlo tree search for goal-oriented dialogue policy planning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7101–7125, Singapore. Association for Computational Linguistics.

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. In *Advances in Neural Information Processing Systems*, volume 37, pages 64735–64772. Curran Associates, Inc.

Tong Zhang, Chen Huang, Yang Deng, Hongru Liang, Jia Liu, Zujie Wen, Wenqiang Lei, and Tat-Seng Chua. 2024b. Strength lies in differences! Improving strategy planning for non-collaborative dialogues via diversified user simulation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 424–444, Miami, Florida, USA. 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*.

Zirui Zhao, Wee Sun Lee, and David Hsu. 2023. Large language models as commonsense knowledge for large-scale task planning. In *Advances in Neural Information Processing Systems*, volume 36, pages 31967–31987. Curran Associates, Inc.

Shenzhe Zhu, Jiao Sun, Yi Nian, Tobin South, Alex Pentland, and Jiaxin Pei. 2025a. The automated but risky game: Modeling agent-to-agent negotiations and transactions in consumer markets. In *First Workshop on Multi-Turn Interactions in Large Language Models*.

Shenzhe Zhu, Jiao Sun, Yi Nian, Tobin South, Alex Pentland, and Jiaxin Pei. 2025b. The automated but risky game: Modeling agent-to-agent negotiations and transactions in consumer markets. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*.

Appendix

This appendix provides supplementary material to support the main paper. Appendix A details the four-phase MCTS procedure used in our strategic reasoning module. Appendix B describes the implementation of the baseline systems compared in our experiments. Appendix C presents the prompt templates, the buyer strategy taxonomy, and the user simulation setup used to construct diverse seller personas. Finally, Appendix D reports the training configurations and data statistics for the analyzer (SFT) and actor (DPO) modules.

We will release our code upon acceptance to facilitate the community's progress on this topic.

A Strategic Reasoning through MCTS: Implementation Details

The implementation of our MCTS is built on the search procedure introduced by GDP-Zero (Yu et al., 2023) and adapt it to the negotiation setting. This section complements the main paper by laying out how MCTS is instantiated in our pipeline. At each turn i, we represent the dialogue context as the history of past acts $s_{tr}^{i} = (a_0, \ldots, a_i)$, and write a^* for the strategy ultimately committed to at that turn. The search proceeds through four phases that are repeated for K iterations; across iterations, we maintain running statistics for each visited node — the action value Q, the state value V, and the visit count N — and use the accumulated statistics to pick the final strategy once the budget is exhausted. The four phases are as follows:

Selection. For the current node s_{tr}, this phase aims to choose a dialogue strategy a to reach a child node. To balance exploration and exploitation, we employ the PUCT function to select the dialogue strategy:

$$\mathrm{PUCT}(s_{tr}, a) = Q(s_{tr}, a) + c_p \cdot P(s_{tr}, a) \cdot \frac{\sqrt{\sum_{a'} N(s_{tr}, a')}}{1 + N(s_{tr}, a)}$$

The algorithm keeps selecting a child node in sequence until a leaf node is reached.

Expansion. Upon reaching a leaf node, we use the LLM as a policy network to obtain a prior distribution over the candidate strategies $\mathcal{A}$. We directly read the LLM's log-probability ℓ_a for each strategy token a conditioned on the current dialogue context, and convert these scores into a prior via a temperature-scaled softmax:

$$P(a \mid s_{tr}) = \frac{\exp(\ell_a/\tau)}{\sum_{a' \in \mathcal{A}} \exp(\ell_{a'}/\tau)}$$

where τ controls how peaked the prior is. The resulting probabilities populate $P(s_{tr}, a)$ in the PUCT formula. Each newly expanded child is initialized with $Q(s_{tr}, \cdot) = Q_0$, a hyperparameter that biases early exploration.

Evaluation. We determine the value of a state based on the likelihood of its dialogue context leading to task success. In our buyer-side negotiation task, the goal is to reach a mutually agreeable price with the seller, so we evaluate a leaf state by estimating the expected profit reward conditioned on the current price context and the seller's likelihood of acceptance. Because the seller is a stochastic LLM, we additionally maintain a small pool of sampled rollouts per node to stabilize this estimate.

Backpropagation. After each search iteration concludes, we perform the following updates:

$$N(s_{tr}, a) \leftarrow N(s_{tr}, a) + 1$$
$$Q(s_{tr}, a) \leftarrow Q(s_{tr}, a) + \Delta Q(s_{tr}, a)$$
$$\Delta Q(s_{tr}, a) = \frac{v(s_{tr}) - Q(s_{tr}, a)}{N(s_{tr}, a)}$$

After all simulations, we select the optimal strategy for the current state s_{tr}^{0} using:

$$a^* = \arg\max_a \; N(s_{tr}^{0}, a)$$

B Implementation of baselines

We provide details on how we implemented the baselines in the main results.

Pro-CoT: Following Deng et al. 2023, we prompt Qwen3-14B to reason explicitly in three sequential steps before generating a response: (1) analyze the current negotiation progress and set a goal, (2) select an appropriate strategy and a dialogue act, and (3) generate a response grounded in the reasoning.

ICL-AIF: Following Fu et al. 2023, we prompt Qwen3-14B to generate three verbal suggestions for the dialogue agent after each completed negotiation. These suggestions are accumulated in a rolling buffer that is capped at 20 and injected into every subsequent turn as in-context guidance.

DMNA: Following Liu et al. 2025, we fine-tune a Qwen3-14B actor model with DPO as the intuitive

module, which generates a stategic response in a single forward pass. We use GPT-4o-mini(OpenAI, 2024) as the reflexion module: at each turn, 3 independent critic instances score the response, and if a majority flag it as poor, a moderator synthesizes their feedback and the actor regenerates with the critique appended as memory. We set the number of critics to 3 and the maximum number of reflexion iterations to 3.

C Details of Prompting

C.1 Prompt Template

We use four prompt types in each domain. All prompts follow a role-playing framing and produce structured outputs that are parsed by regular expression. The CB and A2A-NT domains share the same overall prompt structure but differ in their item-context fields and in whether a seller cost floor is made explicit. Placeholders in curly braces (e.g., `{title}`) are substituted at runtime. The full prompt texts are at Tables 10–16.

- **Buyer Prompt**: generate a single negotiation utterance conditioned on a pre-selected strategy (CB: Table 10; A2A-NT: Table 11).
- **Seller Prompt**: simulate the opponent, conditioned on the seller persona (Appendix C.3) and conversation history (CB: Table 12; A2A-NT: Table 13).
- **Strategy Selection Prompt**: provide the MCTS with log-probabilities over the buyer strategies (Appendix C.2) (CB and A2A-NT: Table 14;).
- **Analyzer Prompt**: update the buyer's belief about the seller after each real negotiation turn, using the SFT-fine-tuned analyzer model (CB: Table 15; A2A-NT: Table 16).

C.2 Details of strategies

We present the strategies used across both datasets, comprising a total of 11 distinct strategies. Detailed descriptions of each strategy are provided in Table 9.

C.3 User Simulation

To stress-test the buyer agent against a diverse opponent population, each simulated seller is assigned a persona drawn from a small grid of traits and styles. We build our prompt upon the prompts introduced in DMNA (Liu et al., 2025), which are based on TRIP (Zhang et al., 2024b) and Dutt et al. 2021. The prompts are located at Table 12 and Table 13.

A persona combines three pieces: a background archetype (e.g., a private seller, a small reseller), one of five *Big Five personality* traits, and one of four *Decision-Making* styles. The assignment is scenario-indexed and deterministic so that the evaluation set is reproducible and the trait/style dimensions remain decorrelated across scenarios. The resistance strategies follow (Dutt et al., 2021) and are incorporated into the prompt.

D Details of Training

We train two LoRA adapters on top of a Qwen3-14B base model: an **analyzer** (SFT) and an **actor** (DPO). Through self-play MCTS, we generated 3,857 preference pairs for DPO training and 10,372 SFT training instances for the analyzer. High-level training settings are summarized in Table 8.

	Analyzer (SFT)	**Actor (DPO)**
Objective	Supervised next-token	DPO preference loss
Learning rate	1×10^{-4}	5×10^{-6}
Effective batch	8	16
Epochs	3	3
DPO β	—	0.1

Table 8: Training settings for the two fine-tuned models. Both share the same LoRA configuration on a 4-bit Qwen3-14B base.

The **analyzer** is trained with SFT to fill in the structured prediction block of the analyzer prompt: given a completed negotiation turn, it predicts the seller's target price, personality, and decision style (plus a cost floor in the A2A domain). Training entries are sampled from negotiations with known ground-truth information.

The **actor** is trained with Direct Preference Optimization (Rafailov et al., 2023). Following DMNA (Liu et al., 2025), preference pairs are mined from MCTS search trees on the training split: for each scenario, the highest-reward strategy realization is treated as the chosen response, and other realizations from the same tree are treated as rejected responses. The choice of $\beta = 0.1$ follows standard practice in preference-based LLM alignment, trading off policy movement against staying close to the base model.

#	Strategy	Description
1	Greetings	Open the conversation casually.
2	Ask a question	Ask about features, condition, or history.
3	Answer a question	Provide information or clarify when asked.
4	Propose the first price	Initiate a price offer or range.
5	Propose a counter price	Counter the seller's offer with a new price.
6	Use comparatives	Reference market prices to justify a lower offer.
7	Confirm information	Ask the seller to confirm details.
8	Affirm confirmation	Affirm the seller's statement or question.
9	Deny confirmation	Push back on the seller's statement.
10	Agree with the proposal	Accept the seller's price.
11	Disagree with a proposal	Reject the seller's price and push lower.

Table 9: Details of the strategies used in both CB and A2A-NT.

The buyer prompt in CB
You are the buyer in a price bargaining game. Now enter the role-playing mode. Your strategy for this turn is: [{strategy_name}]: {strategy_desc}. Please chat with the Seller using a short and natural sentence. You are trying to buy the {title} with a target price of {target_str}. Product description: {description}. You must use the strategy [{strategy_name}] and provide a corresponding response based on the Conversation Context. Please reply with only one short and succinct sentence. Conversation Context: {history}. Seller's last message: {seller_dialogue} [${seller_price}]. Please output the appropriate and high-quality response and format your answer as '[{strategy_name}]response'. Your Response:

Table 10: The prompt design of the buyer in CB.

The buyer prompt in A2A-NT
You are a buyer negotiating to purchase a {product_type}. Now enter the role-playing mode. Your strategy for this turn is: [{strategy_name}]: {strategy_desc}. You are trying to buy {name}. Product features: {features}. Listed at ${listing_price}. Your target price is {target_str}. Use the strategy [{strategy_name}] and reply with only one short and succinct sentence. Conversation Context: {history}. Seller's last message: {seller_dialogue} [${seller_price}]. Format your answer as '[{strategy_name}]response'. Your Response:

Table 11: The prompt design of the buyer in A2A-NT.

The seller prompt in CB
You are a seller in a price bargaining game. Now enter the role-playing mode. Your persona: {seller_persona}. You must follow these instructions during the conversation: (1) Your utterances and bargaining behavior must strictly follow your persona; vary your wording and avoid repeating yourself verbatim; (2) You may decide to change your target price flexibly based on your persona and the conversation. You may draw on the following conversation strategies: 1. "Source Derogation": attack the other party or question the item 2. "Counter Argument": refute a previous claim or justify a new claim with a non-personal argument 3. "Personal Choice": give a personal reason for disagreeing or conditionally agreeing 4. "Information Inquiry": request clarification or additional information 5. "Self Pity": give a sympathetic reason for disagreeing 6. "Hesitance": stall to encourage a better offer) 7. "Self-assertion": assert a new claim with finality 8. "Others": do not explicitly foil the negotiation. You are trying to sell the {title} with an initial price of ${listing_price}. Product description: {description}. Reply with only one short and succinct sentence.

Table 12: The prompt design of the seller in CB.

The seller prompt in A2A-NT
You are a {product_type} seller. Now enter the role-playing mode. Your persona: {seller_persona}. You must follow these instructions during the conversation: (1) Your utterances and bargaining behavior must strictly follow your persona; vary your wording and avoid repeating yourself verbatim; (2) You may decide to change your target price flexibly based on your persona and the conversation. You may draw on the following conversation strategies: 1. "Source Derogation": attack the other party or question the item 2. "Counter Argument": refute a previous claim or justify a new claim with a non-personal argument 3. "Personal Choice": give a personal reason for disagreeing or conditionally agreeing 4. "Information Inquiry": request clarification or additional information 5. "Self Pity": give a sympathetic reason for disagreeing 6. "Hesitance": stall to encourage a better offer) 7. "Self-assertion": assert a new claim with finality 8. "Others": do not explicitly foil the negotiation. You are selling {name}. Product features: {features}. Listed at ${listing_price}. Your cost floor is ${seller_target} — do not accept below this. Reply with only one short and succinct sentence.

Table 13: The prompt design of the seller in A2A-NT.

The strategy-selection prompt in CB and A2A-NT
You are a buyer negotiating to purchase {title} (listed at ${listing_price}). Your target price is {target_str}. Conversation so far: {history}. The available strategies are: ${strategy_list} Select the MOST APPROPRIATE strategy for this moment in the negotiation. Consider: how far along is the negotiation; what did the seller last say; which action would best advance your goal of getting a lower price. Respond with the strategy name in brackets, e.g., [Propose a counter price] or [Disagree with a proposal]. Strategy:

Table 14: The prompt design of the strategy-selection module in CB and A2A-NT.

The analyzer prompt in CB
Given the following price negotiation context, predict the seller's target price and next action. {negotiation_history}. Buyer's Current Offer: ${buyer_price}. Buyer: {buyer_dialogue}. Seller Response: {seller_response}. Task: based on the seller's response and the negotiation context, predict the seller's target price and likely next action. Please format your response with the following fields, one per line: Seller Target Price, Seller Action, Seller Counter Price, Big Five Personality, Decision Making Style.

Table 15: The prompt design of the analyzer in CB.

The analyzer prompt in A2A-NT
Given the following price negotiation context, predict the seller's target price and next action. {negotiation_history}. Buyer's Current Offer: ${buyer_price}. Buyer: {buyer_dialogue}. Seller Response: {seller_response}. Task: based on the seller's response and the negotiation context, predict the seller's target price and likely next action. Please format your response with the following fields, one per line: Seller Target Price, Seller Cost Floor, Seller Action, Seller Counter Price, Big Five Personality, Decision Making Style.

Table 16: The prompt design of the analyzer in A2A-NT.