One Planner To Guide Them All ! Learning Adaptive Conversational Planners for Goal-oriented Dialogues

Huy Dao
Singapore Management University
qh.dao.2023@phdcs.smu.edu.sg

Lizi Liao
Singapore Management University
lzliao@smu.edu.sg

Abstract

Goal-oriented dialogues, such as recommendation and negotiation, often require balancing multiple conflicting objectives. Conventional approaches typically train separate policies for each predefined objective trade-off, which is computationally costly and scales poorly. In this work, we pursue a single dialogue policy that can dynamically adapt to varying objective preferences at inference time **without retraining**. This raises several challenges in terms of both **(1) optimization strategy** and **(2) knowledge utilization**. To address these, we propose a novel policy learning framework, **P**reference **A**daptive **D**ialogue **P**olicy **P**lanner (PADPP), for multi-objective goal-oriented dialogues. Specifically, to tackle the former, we introduce a novel optimization scheme, which leverages information gained from training the model on previously updated objective weights, accelerating the learning capability on new weight settings. To address the latter, we utilize Generalized Policy Improvement (GPI) to ensure the effectiveness of leveraged knowledge. Experimental results demonstrate that PADPP achieves superior adaptability and performance compared to state-of-the-art approaches, offering a scalable and flexible solution for multi-objective, goal-oriented dialogues [1].

1 Introduction

Balancing multiple, potentially conflicting objectives is a core challenge in goal-oriented dialogue systems, particularly in domains like negotiation (He et al., 2018; Deng et al., 2023b; Zhang et al., 2024) and recommendation (Liu et al., 2020b; Zhang et al., 2021; Liu et al., 2021; Dao et al., 2023). For example, a negotiation agent may try to maximize profit while maintaining fairness (He et al., 2018), whereas a recommendation system might seek to improve user satisfaction without sacrificing the quality of its suggestions (Liu et al., 2021). Hence, effective dialogue strategies depend on the relative importance assigned to objectives, which can be determined by either the system designers' preferences or their domain expertise. For instance, as illustrated in Figure 1, varying the objective priorities results in distinct strategies and conversation outcomes. Moreover, in real-world applications, such priorities are dynamic and can shift upon specific contexts, necessitating fast and efficient adaptation of dialogue systems to changing objective preferences.

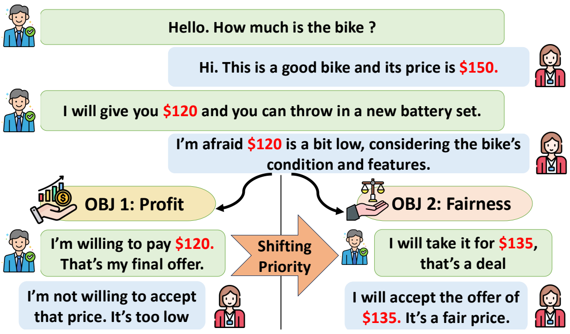

Hello. How much is the bike ?
Hi. This is a good bike and its price is $150.
I will give you $120 and you can throw in a new battery set.
I'm afraid $120 is a bit low, considering the bike's condition and features.
OBJ 1: Profit
OBJ 2: Fairness
I'm willing to pay $120. That's my final offer.
Shifting Priority
I will take it for $135, that's a deal
I'm not willing to accept that price. It's too low
I will accept the offer of $135. It's a fair price.

Figure 1: In goal-oriented dialogues (*e.g.* negotiation), shifting objective priorities shape distinct strategies, requiring a planner to adapt quickly.

However, existing approaches struggle to address the dynamic nature of multi-objective optimization in dialogues. Some methods lack a dedicated training process for optimizing objectives of interest (Deng et al., 2023a), while others face significant computational overhead due to costly simulations (Yu et al., 2023; Dao et al., 2024b). Recent advances have introduced plug-in planners for LLM-based dialogue agents (Deng et al., 2023b; He et al., 2024a; Zhang et al., 2024), which optimize smaller models, such as RoBERTa (Liu et al., 2019), using Reinforcement Learning (RL) to improve both planning capability and computational efficiency. Despite being effective for fixed objective priorities, such methods are static planners, requiring expensive retraining for new preferences.

[1] Code: https://github.com/huyquangdao/PADPP

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 22092–22116
November 4-9, 2025 ©2025 Association for Computational Linguistics

Recently, Multi-Objective Reinforcement Learning (MORL) has emerged as a prominent framework for optimizing multiple objectives simultaneously (Abels et al., 2019; Yang et al., 2019; Hayes et al., 2022). Unlike the conventional RL approach, MORL aims to produce solutions that balance the trade-offs between objectives, making it appealing for dialogue scenarios with competing goals. Yet two main challenges arise when applying MORL to dialogue planners: (1) **Optimization Strategy**, where training from scratch for each new preference is both slow and sample-inefficient, and (2) **Knowledge Utilization**, where it remains unclear how best to leverage previously learned solutions.

To address these challenges, we propose PADPP, a **P**reference **A**daptive **D**ialogue **P**olicy **P**lanner, which enhances Double Deep Q-Networks (DDQN) (Hasselt et al., 2016) with a knowledge reuse mechanism. Specifically, during the training phase, given an objective preference, PADPP optimizes an additional objective function, aiming to distill knowledge from a *teacher* policy that is derived from the set of previously learned solutions. This auxiliary distillation step speeds up convergence by reusing insights acquired in past training iterations. Furthermore, we propose to instantiate the *teacher* with *Generalized Policy Improvement* (GPI) (Barreto et al., 2017), theoretically ensuring that the induced policy is no worse than any single policy in the set. Crucially, after training, PADPP can seamlessly handle arbitrary objective preferences *without* retraining, facilitating efficient and flexible deployment in multi-objective, goal-oriented dialogues. Experimental results on two published datasets, namely **Craigslist Bargain** (He et al., 2018) and **DuRecDial 2.0** (Liu et al., 2021), demonstrate the superiority and adaptability of our proposed PADPP against SOTA approaches on changing objective priority situations.

In summary, our contributions are threefold:

- To the best of our knowledge, we are the first to introduce an adaptive policy planner for multi-objective, goal-oriented dialogues.
- We propose a novel policy learning method, named PADPP, aiming to enhance DDQN with a knowledge reuse mechanism.
- We conduct extensive experiments on two published benchmarks. Empirical results demonstrate the superiority and adaptability of our methods against state-of-the-art approaches.

2 Related Work

2.1 Goal-oriented Dialogue Systems

Recent works on goal-oriented dialogue systems have largely focused on tasks like negotiation and recommendation, requiring these systems to manage multiple, sometimes conflicting objectives (Liu et al., 2020b; Zhang et al., 2021; Liu et al., 2021; Wang et al., 2022; Deng et al., 2023c; Wang et al., 2023; Dao et al., 2023). Approaches based on LLMs often rely on prompt engineering (*e.g.*, Chain-of-Thought) to enhance planning (Deng et al., 2023a), but these methods lack explicit training processes to optimize arbitrary objective combinations. Other efforts employ simulation-based algorithms such as Monte-Carlo Tree Search (MCTS) (Yu et al., 2023; He et al., 2024a; Dao et al., 2024b), which improve planning at the cost of heavy computational overhead during inference.

To mitigate these limitations, a surge of works has explored plug-and-play policy planners trained via RL to optimize fixed objective combinations (Deng et al., 2023b; He et al., 2024a; Zhang et al., 2024). For example, Deng et al. (2023b) utilized REINFORCE (Williams, 1992) for static objective preferences, while He et al. (2024a) proposed a dual-process RL approach combining offline RL pre-training with MCTS self-play. Zhang et al. (2024) extended this framework by integrating user-awareness information and diverse simulators. Despite achieving promising performance, these methods remain *static*, requiring retraining when objective priorities shift.

In contrast, we introduce PADPP, a novel, *adaptive* policy planning approach that directly accommodates varying objectives. To our knowledge, this is the first method to target multi-objective adaptability in goal-oriented dialogue systems.

2.2 Multi-objective Reinforcement Learning

Recently, MORL (Abels et al., 2019; Yang et al., 2019; Hayes et al., 2022; Chen et al., 2023; Hwang et al., 2024) has gained traction for tasks that demand balancing multiple, often conflicting goals. For example, Hwang et al. (2024) employed MORL to train an adaptable navigation agent by incorporating diverse forms of interaction, including human demonstrations, trajectory comparisons, and natural language instructions. In another context, Chen et al. (2023) utilized MORL to fine-tune a Variational Autoencoder (VAE) model (Kingma and Welling, 2014), effectively balancing fidelity and diversity in conditional text generation. Inspired by these efforts, we aim to explore the capability of MORL in learning goal-oriented dialogue policies where conflicting objectives frequently arise, and their relative importance often shifts dynamically in real-world applications. We identify two primary challenges in applying MORL to this domain: (1) Developing effective optimization strategies and (2) Leveraging existing knowledge effectively. To address these challenges, we propose the PADPP method, aiming to enhance the standard DDQN method with a knowledge reuse mechanism.

3 Preliminaries

3.1 Problem Formulation

We formalize dialogue policy planning as a Multi-objective Markov Decision Process (MOMDP), defined as $M \triangleq (\mathcal{S}, \mathcal{A}, T, \mathbf{r}, \mu, \gamma)$ where $\mathcal{S}, \mathcal{A}$ are the state and action spaces, respectively. $T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ defines a state transition function, and $\mathbf{r} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^d$ is a vector-valued reward function. $\mu \in \Delta_{|\mathcal{S}|}$ is a probability distribution over initial states and $\gamma \in [0, 1]$ is a discount factor. In goal-oriented dialogues, we are often interested in d different and possibly conflicting objectives $o_i, i = 1..d$ where each objective o_i associates with a reward signal r_i. Given a vector $\mathbf{w} \in \Delta_d$ (*i.e.*, $\sum_{i=1}^{d} \mathbf{w}_i = 1$) indicating our preference over these objectives, a scalar reward signal $r_{\mathbf{w}}$ could be computed as a weighted sum of reward signals and their corresponding objective weights (*i.e.*, $r_{\mathbf{w}} = \sum_{i=1}^{d} \mathbf{w}_i r_i$). Our goal is to induce a policy function $\pi : \mathcal{S} \times \Delta_d \rightarrow \mathcal{A}$ which maps a state $s \in S$ (s could be our dialogue history) and a preference weight $\mathbf{w} \in \Delta_d$ to a specific action $a \in \mathcal{A}$ that maximizes the scalarized accumulated rewards. For convenience, we denote the policy for a specific $\mathbf{w}$ as $\pi_{\mathbf{w}}$. Correspondingly, we denote by $\mathbf{Q}^{\pi_{\mathbf{w}}}(s, a) \triangleq \mathbb{E}_{\pi_{\mathbf{w}}}\left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}(s_t, a_t) \mid s_0 = s, a_0 = a\right]$ and $\mathbf{V}^{\pi_{\mathbf{w}}}(s) \triangleq \mathbb{E}_{\pi_{\mathbf{w}}}\left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}(s_t, a_t) \mid s_0 = s\right]$ as the vector-valued action and state value functions of $\pi_{\mathbf{w}}$, respectively. Finally, the value function of a policy $\pi_{\mathbf{w}}$ can be computed as $\mathbf{V}^{\pi_{\mathbf{w}}} = \mathbb{E}_{s_0 \sim \mu}\left[\mathbf{V}^{\pi_{\mathbf{w}}}(s_0)\right]$. For notation convenience, we refer to $\mathbf{r}(s, a)$ as $\mathbf{r}$.

3.2 Multi-objective Dialogue Enviroments

In this work, we introduce two interactive multi-objective dialogue environments—*negotiation* (He et al., 2018) and *recommendation* (Liu et al., 2021). In the negotiation scenario, an agent bargains with a simulated seller to reach an agreement, optimizing three objectives: price gain (r_{gain}), fairness (r_{fair}), and deal rate (r_{deal}). In the recommendation scenario, the agent proposes specific items to a simulated user, balancing user sentiment (r_{user}) and item frequency (r_{item}). Each conversation unfolds over a pre-defined number of turns. Following Deng et al. (2023b), we employ LLMs (Liu et al., 2023) to instantiate user simulators. Details on reward calculation and prompting strategies can be found in the A.8 and A.14.

3.3 Double Deep Q-Networks (DDQN)

On-policy RL methods(*e.g.* PPO (Schulman et al., 2017)) have demonstrated outstanding performance in single-objective settings, yet they suffer from sample inefficiency in MORL (Hayes et al., 2022). Therefore, in this work, we establish PADPP based on DDQN (Hasselt et al., 2016) - an off-policy RL approach commonly applied for learning dialogue policies (Wang et al., 2020; Zhao et al., 2021). Specifically, we parameterize the standard state-action value function for a particular policy $\pi_{\mathbf{w}}$ by a DQN $Q^{\pi_{\mathbf{w}}}(s, a; \theta_{\mathbf{w}}) \in \mathbb{R}$ where $\theta_{\mathbf{w}}$ are the corresponding parameters. To train DDQN, we optimize the parameters $\theta_{\mathbf{w}}$ by minimizing the TD loss function, defined as follows:

$$l_{\mathbf{w}}(\theta_{\mathbf{w}}) = \mathbb{E}_{(s,a,\mathbf{r},s') \sim \mathcal{B}}\left[(y_{\mathbf{w}} - Q^{\pi_{\mathbf{w}}}(s, a; \theta_{\mathbf{w}}))^2\right],$$

where $y_{\mathbf{w}} \triangleq$

$$\begin{cases} \mathbf{w}^{\mathrm{T}}\mathbf{r}, \text{if } s' \text{ is the terminal state}, \\ \mathbf{w}^{\mathrm{T}}\mathbf{r} + \gamma \max_{a \in \mathcal{A}} Q^{\pi_{\mathbf{w}}}(s', a; \theta_{\mathbf{w}}^{-}), \text{others}, \end{cases}$$

where $\mathcal{B}$ is a replay buffer and $\theta_{\mathbf{w}}^{-}$ are delayed parameters. However, naively applying DDQN approach for MORL suffers from both inefficiency and scalability issues. Specifically, suppose we have learned $Q^{\pi_{\mathbf{w}'}}(s, a; \theta_{\mathbf{w}'})$ for some particular $\mathbf{w}'$, this knowledge could not be used to guide the learning of $Q^{\pi_{\mathbf{w}}}(s, a; \theta_{\mathbf{w}})$. Furthermore, one might need to store multiple instances of $Q^{\pi_{\mathbf{w}'}}(s, a; \theta_{\mathbf{w}'})$ for different preferences $\mathbf{w}'$.

4 The PADPP Method

In the following sections, we introduce our PADPP for learning adaptive and multi-objective dialogue strategies, which is diverged from existing approaches at both training and inference. Moreover, we also provide theoretical justifications to support its effectiveness in multi-objective settings.

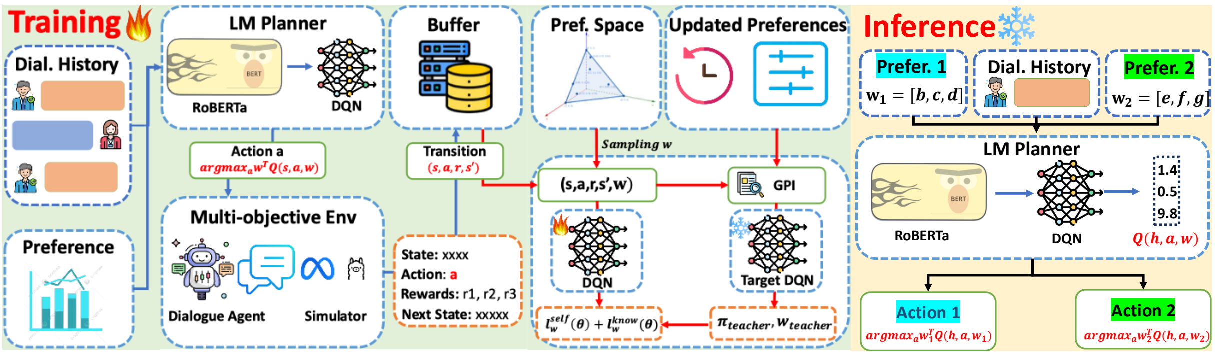


Figure 2: The illustration of our proposed PADPP method for multi-objective, goal-oriented dialogues. Specifically, the training and inference phases are described in Section 4.2 and 4.3, respectively. In the figure, blue arrows indicate the execution phase while red arrows depict the optimization phase in our training, detailed in Algorithm 1.

4.1 Framework Overview

We show an overview of our PADPP method in Figure 2. Following (Deng et al., 2023b; Zhang et al., 2024), we establish PADPP, using a small LM (*e.g.*, RoBERTa (Liu et al., 2019)) and a DQN, as a plug-in policy planner. Unlike prior methods that optimize a single, fixed objective combination, PADPP dynamically adjusts objective weights at both training and inference. To avoid strong assumptions on the underlying distribution of preference weights, PADPP samples preference weights randomly during training. At inference time, given a user-defined weight vector, PADPP outputs an action optimized for the specified trade-off.

4.2 PADPP's Training Procedure

Generally, the core idea in PADPP's training procedure is to learn $\mathbf{Q}^{\pi_{\mathbf{w}}}(s,a)$ for $\pi_{\mathbf{w}}$ by leveraging the knowledge gained from learning $\mathbf{Q}^{\pi_{\mathbf{w}'}}(s,a)$ on other past objective configurations $\mathbf{w}'$.

4.2.1 DDQN with Knowledge Reuse

The detailed training algorithm is described in Algorithm 1. For a compact representation, inspired by Universal Value Function Approximators (UVFA) (Schaul et al., 2015), we parameterize $\mathbf{Q}^{\pi_{\mathbf{w}}}(s,a)$ using a DQN defined as $\mathbf{Q}(s,a,\mathbf{w};\theta) \in \mathbb{R}^d$ where θ are parameters. Similar to standard DDQN, our algorithm consists of two phases, including policy execution and optimization. In the former stage, we utilize the epsilon-greedy and store experience $(s,a,\mathbf{r},s')$ into a buffer $\mathcal{B}$. In the latter phase, we randomly sample an objective preference $\mathbf{w} \sim \Delta_{|d|}$ and jointly optimize the Q function using two different objective functions.

Self-Learning. In this step, we learn θ by following the current preference $\mathbf{w}$. Given a transition $(s,a,\mathbf{r},s') \sim \mathcal{B}$, we scalarize the reward vector $\mathbf{r}$ using $\mathbf{w}$ and computing a TD error function, formulated as follows:

$$l^{\text{self}}_{\mathbf{w}}(\theta) = \mathbb{E}_{(s,a,\mathbf{r},s')}\left[(y^{\text{self}}_{\mathbf{w}} - \mathbf{w}^{\mathrm{T}}\mathbf{Q}(s,a,\mathbf{w};\theta))^2\right],$$

where $y^{\text{self}}_{\mathbf{w}} \triangleq$

$$\begin{cases} \mathbf{w}^{\mathrm{T}}\mathbf{r}, \text{if } s' \text{ is the terminal state,} \\ \mathbf{w}^{\mathrm{T}}\mathbf{r} + \gamma \max\limits_{a \in \mathcal{A}} \mathbf{w}^{\mathrm{T}}\mathbf{Q}(s',a,\mathbf{w};\theta^-), \text{others.} \end{cases} \quad (1)$$

Afterward, we append the sampled preference weight w to a memory buffer $\mathcal{W}$, which stores previously updated preferences.

Knowledge Reuse. Intuitively, to solve a new task, one might need to leverage the knowledge of solving the other tasks in the past. Hence, it is desired to leverage useful knowledge, such as learned state-action value function $\mathbf{Q}^{\pi_{\mathbf{w}_1}}(s,a)$ or state value function $\mathbf{V}^{\pi_{\mathbf{w}_1}}(s)$, for $\mathbf{w}_1$ (*e.g.*$[1.0, 0.0]$) to enhance the learning process on a new preference $\mathbf{w}_2$ (*e.g.*$[0.9, 0.1]$). Formally, given the set of past updated preferences $\mathcal{W}$, we aim to distill useful knowledge from this set to accelerate the learning on $\mathbf{w}$. Formally, we optimize an additional auxiliary objective function, defined as follows:

$$l^{\text{know}}_{\mathbf{w}}(\theta) = \mathbb{E}_{(s,a,\mathbf{r},s')}\left[||\mathbf{y}^{\text{know}}_{\mathbf{w}} - \mathbf{Q}(s,a,\mathbf{w};\theta)||_2^2\right],$$

where $\mathbf{y}^{\text{know}}_{\mathbf{w}} \triangleq$

$$\begin{cases} \mathbf{r}, \text{if s' is the terminal state,} \\ \mathbf{r} + \gamma\mathbf{Q}(s',\pi_{\text{teacher}},\mathbf{w}_{\text{teacher}};\theta^-), \text{others,} \end{cases} \quad (2)$$

where $\pi_{\text{teacher}} \in \mathcal{A}$, $\mathbf{w}_{\text{teacher}} \in \mathcal{W}$ are arguments for the state-action value function, which are induced by using $\mathbf{w}$, $\mathbf{Q}(s',.,.;\theta^-)$ and the set $\mathcal{W}$. Finally, for model training, we optimize a weighted combination of $l^{\text{self}}_{\mathbf{w}}(\theta)$ and $l^{\text{know}}_{\mathbf{w}}(\theta)$, defined as follows:

$$l_{\mathbf{w}}(\theta) = \alpha * l^{\text{know}}_{\mathbf{w}}(\theta) + (1-\alpha) * l^{\text{self}}_{\mathbf{w}}(\theta),$$

where α is a hyperparameter to balance these two objectives and can be adjusted during the training.

4.2.2 Teacher Selection Mechanism

Intuitively, a good teacher might guide the learning process better. Hence, the instantiation of π_{teacher} and $\mathbf{w}_{\text{teacher}}$ is critical and has a significant impact on the model performance. For instance, one could instantiate $\mathbf{w}_{\text{teacher}}$ using $\mathbf{w}_{\text{MinDist}} \in \mathcal{W}$, which is the closest one to the current $\mathbf{w}$ (*e.g.* $\mathbf{w}_{\text{MinDist}} = \arg\min_{\mathbf{w}_i \in \mathcal{W}} d(\mathbf{w}, \mathbf{w}_i)$) where d is an arbitrary distance function. In this work, we propose to instantiate those two by leveraging GPI (Barreto et al., 2017), formulated as follows:

$$\pi_{\text{teacher}}, \mathbf{w}_{\text{teacher}} \in \arg\max_{a, \mathbf{w}_i} \max_{a \in \mathcal{A}, \mathbf{w}_i \in \mathcal{W}} \mathbf{w}^{\mathrm{T}} \mathbf{Q}(s, a, \mathbf{w}_i; \theta).$$

Additionally, the following theorem demonstrates that under general conditions, our GPI-based knowledge reusing could extract *the best teacher* from the set of previously updated preferences $\mathcal{W}$.

***Theorem* 1** *Given a set of policies $\Pi_{\mathcal{W}} = \{\pi_{\mathbf{w_i}}\}_{i=1}^{|\mathcal{W}|}$, their corresponding state-action value functions $\mathbf{Q}^{\pi_{\mathbf{w_i}}}(s, a)$ and a new preference $\mathbf{w}$, denote by $\pi_{\mathbf{w}}^{gpi}$ the GPI policy induced from set $\Pi_{\mathcal{W}}$, such that:*

$$\pi_{\mathbf{w}}^{gpi} \in \arg\max_{a \in \mathcal{A}} \max_{\mathbf{w}_i \in \mathcal{W}} \mathbf{w}^T \mathbf{Q}^{\pi_{\mathbf{w}_i}}(s, a),$$

then we have:

$$\mathbf{w}^T \mathbf{V}^{\pi_{\mathbf{w}}^{gpi}} \geq \mathbf{w}^T \mathbf{V}^{\pi_{\mathbf{w}_i}}, \forall \mathbf{w}_i \in \mathcal{W},$$

For completeness, we provide detailed discussions regarding our knowledge reusing approach and potential instantiations of $\pi_{\text{teacher}}, \mathbf{w}_{\text{teacher}}$ in Appendix A.1. Finally, the proofs of Theorem 1 can be found in Appendix A.2.

4.3 Preference Adaptive Dialogue Planning

In addition to multi-objective optimization, a key innovation of our model lies in its adaptability to dynamic shifts in objective preferences without retraining the model. Specifically, during inference, one could first specify an arbitrary vector $\mathbf{w}_{\text{infer}} \in \Delta_d$ representing his/her preference over objectives. Given a dialogue history h, the corresponding action could be computed using the following formulation:

$$a_{\text{infer}} = \arg\max_{a \in \mathcal{A}} \mathbf{w}_{\text{infer}}^{\mathrm{T}} \mathbf{Q}(h, a, \mathbf{w}_{\text{infer}}; \theta), \quad (3)$$

where $\mathbf{w}_{\text{infer}}$ can be dynamically adjusted at either turn-level or dialogue-level. This flexibility enhances computational efficiency and enables adaptive strategies that cater to the evolving progress of the conversation with diverse end users.

Algorithm 1 PADPP's Training Algorithm

```
Input: epsilon $\epsilon$, discount factor $\gamma$, balancing parameter $\alpha$, network parameters $\theta$;
Output: Learned network parameters $\hat{\theta}$;
Initialize Buffer $\mathcal{B}$, Memory Buffer $\mathcal{W}$;
for episode $\leftarrow 1$ to $M$ do
    randomly sample $\mathbf{w} \sim \Delta_d$;          ▷ Execution Phase;
    $s \leftarrow s_0$;
    while $s$ is not terminal do
        Select an action $a$ using epsilon-greedy:
        $a = \begin{cases} \text{random action in } \mathcal{A}, & \epsilon, \\ \max_{a \in \mathcal{A}} \mathbf{w}^{\mathrm{T}} \mathbf{Q}(s, a, \mathbf{w}; \theta), & 1 - \epsilon, \end{cases}$
        Observe vector reward $\mathbf{r}$ and next state $s'$;
        Store transition $(s, a, \mathbf{r}, s')$ to $\mathcal{B}$;
        Update the state $s \leftarrow s'$;
    end while
    if update then          ▷ Optimization Phase;
        Sample a preference weight $\mathbf{w} \sim \Delta_d$;
        Compute TD loss: $l_{\mathbf{w}}^{\text{self}}(\theta) = \mathbb{E}_{(s,a,\mathbf{r},s')} \left[ (y_{\mathbf{w}}^{\text{self}} - \mathbf{w}^{\mathrm{T}} \mathbf{Q}(s, a, \mathbf{w}; \theta))^2 \right]$ (Eq. 1)
        Store $\mathbf{w}$ to $\mathcal{W}$ if $\mathbf{w} \notin \mathcal{W}$;
        Sample a set of updated preferences from $\mathcal{W}$;
        Compute auxiliary objective: $l_{\mathbf{w}}^{\text{know}}(\theta) = \mathbb{E}_{(s,a,\mathbf{r},s')} \left[ ||\mathbf{y}_{\mathbf{w}}^{\text{know}} - \mathbf{Q}(s, a, \mathbf{w}; \theta)||_2^2 \right]$ (Eq. 2)
        Compute $l_{\mathbf{w}}(\theta) = \alpha * l_{\mathbf{w}}^{\text{know}}(\theta) + (1 - \alpha) l_{\mathbf{w}}^{\text{self}}(\theta)$;
        Update the parameters $\theta$ using gradient $\nabla_\theta l_w(\theta)$;
    end if
end for
Return: Network parameters $\theta$;
```

5 Experiments

5.1 Datasets

We conduct experiments on two published datasets, including **Craigslist Bargain** (He et al., 2018) and **DuRecDial 2.0** (Liu et al., 2021), which have been utilized for evaluating state-of-the-art dialogue policies. Specifically, Craigslist Bargain comprises dialogues between buyers and sellers negotiating the price of a product. DuRecDial 2.0 is a recommendation dialogue dataset featuring conversations across various domains, namely movies, music, and points of interest (POI), where each conversation is linked to a target item, and the objective is to successfully recommend this item to the user. In this work, we utilize the standard data splits for training, validation, and testing. Detailed statistics are shown in Table 1 and Appendix A.12.

	DuRecDial 2.0	**Craislist Bargain**
# dialogues	10K	3666
# actions	13	11
# objectives	2	3
# cases	425/272/346	3290/188/188
domains	Movie/Music/POI	-

Table 1: The detailed statistics of datasets.

5.2 Baselines

We compare our proposed method against both traditional RL and state-of-the-art dialogue policy models. For the RL baseline, we include **DDQN** (Hasselt et al., 2016). Among state-of-the-art dialogue policy models, we assess PADPP alongside plug-in policy planners **PPDPP** (Deng et al., 2023b), **DPDP** (He et al., 2024a), and **TRIP** (Zhang et al., 2024). In addition, we conduct ablation studies on **PADPP w/o Know** (excluding knowledge reuse) and variants using alternate teacher-selection strategies (*i.e.* **PADPP-Min Dist**). Implementation details for PADPP and all baselines appear in Appendices A.10 and A.11, respectively.

5.3 Evaluation Metrics

Following prior works, we report the Success Rate (**SR**) and Average Conversation Turn (**Avg. Turn**). We also report the average cumulative rewards (*i.e.*, $r_{\textbf{obj}}$). For negotiation, we present $r_{\textbf{gain}}$, $r_{\textbf{fair}}$, and $r_{\textbf{deal}}$. For recommendation, we show $r_{\textbf{user}}$ and $r_{\textbf{item}}$. Similar to (Hwang et al., 2024), we analyze model performance on changing objective priorities. For negotiation, we report results for three scenarios with weights set as $\mathbf{w}_{\text{infer}} = \mathbf{w}_{\text{obj}} = [\mathbb{1}_{\text{obj}=\textbf{gain}}, \mathbb{1}_{\text{obj}=\textbf{fair}}, \mathbb{1}_{\text{obj}=\textbf{deal}}]$ ($\mathbb{1}$ is the indicator function) where obj is one of considered objectives. For recommendation, we report the performance on two scenarios, namely $\mathbf{w}_{\text{infer}} = \mathbf{w}_{\text{obj}} = [\mathbb{1}_{\text{obj}=\textbf{user}}, \mathbb{1}_{\text{obj}=\textbf{item}}]$. Lastly, we also report the results under a uniform weight (*i.e.*, $\mathbf{w}_{\text{infer}} = \frac{1}{d}\mathbf{1}_d$ where $\mathbf{1}_d$ is a d-dimensional vector of ones). For human evaluation, we invite two annotators to score 20 dialogues across three dimensions in negotiation: **Deal Achievement**, **Negotiation Equity**, and **Buyer's Benefit** (detailed instructions can be found in Appendix A.15) and present the win-lose rate of PADPP compared to baselines. Lastly, we include Kappa statistics (McHugh, 2012) to assess the inter-annotator agreement score. We run all experiments on 3 different random seeds and average the results to get the final performance.

5.4 Experimental Results

Negotiation Dialogues. Table 2 shows how PADPP adapts to shifting objective weights in negotiation tasks. Crucially, PADPP requires only a single training session, whereas baselines must be retrained whenever priorities change, underscoring its core advantage in flexibility and efficiency. Under uniform weighting, PADPP outperforms other methods in 5 of 6 metrics, indicating its strong ability to balance multiple conflicting objectives—specifically, it manages the trade-offs between *Price Gain*, *Fairness*, and *Deal Rate* more effectively than baselines that often over-optimize a single goal. When objective weights shift to prioritize *Price Gain*, *Fairness*, or *Deal Rate*, PADPP consistently maintains top or near-top performance (*e.g.*, $r_{\text{fair}} = 0.368^*$, $r_{\text{deal}} = 0.165^*$), again demonstrating superior adaptability. This robustness highlights PADPP's capacity to handle evolving objective trade-offs in real-world bargaining scenarios, discussed further in Section 6.1.

Model	Re-Train	SR	Avg. Turn	Avg. Cumulated Rewards		
				$r_{\textbf{gain}}$	$r_{\textbf{fair}}$	$r_{\textbf{deal}}$
Uniform						
DDQN$_{\text{uni}}$	✓	0.194	9.911	0.707	0.065	0.060
PPDPP$_{\text{uni}}$	✓	0.128	9.874	0.525	0.149	0.046
DPDP$_{\text{uni}}$	✓	0.123	9.847	0.650	0.077	0.049
TRIP$_{\text{uni}}$	✓	0.129	9.911	**0.739**	0.051	0.042
PADPP	✗	**0.427**	**9.638**	0.622	**0.287**	**0.142**
obj = **Price Gain**						
DDQN$_{\text{gain}}$	✓	0.063	9.994	0.952	-	-
PPDPP$_{\text{gain}}$	✓	**0.183**	**9.857**	0.761	-	-
DPDP$_{\text{gain}}$	✓	0.041	9.921	**0.970***	-	-
TRIP$_{\text{gain}}$	✓	0.179	9.899	0.863	-	-
PADPP	✗	0.085	9.898	0.944	-	-
obj = **Fairness**						
DDQN$_{\text{fair}}$	✓	0.252	9.893	-	0.267	-
PPDPP$_{\text{fair}}$	✓	0.201	**9.736**	-	0.116	-
DPDP$_{\text{fair}}$	✓	0.087	9.889	-	0.228	-
TRIP$_{\text{fair}}$	✓	0.126	9.914	-	0.059	-
PADPP	✗	**0.281**	9.792	-	**0.368***	-
obj = **Deal Rate**						
DDQN$_{\text{deal}}$	✓	0.362	9.748	-	-	0.111
PPDPP$_{\text{deal}}$	✓	0.256	9.805	-	-	0.081
DPDP$_{\text{deal}}$	✓	0.036	9.940	-	-	0.017
TRIP$_{\text{deal}}$	✓	0.224	9.878	-	-	0.065
PADPP	✗	**0.489***	**9.531**	-	-	**0.165***

Table 2: Empirical results on the **Craigslist Bargain** dataset. Except for *uniform*, * indicates the best performance on the corresponding considered objective.

Recommendation Dialogues. Table 3 summarizes results on DuRecDial 2.0. Given the dataset's breadth (Movie, Music, and POI domains), we train and evaluate on each domain separately, then report averages (per-domain results are in the A.5). Notably, PADPP requires only a single training pass for multiple objective priorities, in contrast to baselines that train separate models for each setting. Under uniform objective weights, PADPP demonstrates strong performance in the SR metric, indicating its ability to balance *User Sentiment* and *Item Frequency*—two often conflicting objectives. This tension is evident among baselines: emphasizing frequent recommendations can degrade user

Model	Re-Train	SR	Avg. Turn	Avg. Cumulated Rewards	
				r_{user}	r_{item}
Uniform					
DDQN$_{\text{uni}}$	✓	0.288	10.000	1.016	1.272
PPDPP$_{\text{uni}}$	✓	0.247	10.000	0.481	1.365
DPDP$_{\text{uni}}$	✓	0.197	10.000	0.506	1.924
TRIP$_{\text{uni}}$	✓	0.273	10.000	**2.679**	0.846
PADPP	✗	**0.505**	10.000	2.232	**2.206**
obj = **User Sentiment**					
DDQN$_{\text{user}}$	✓	0.047	10.000	1.187	-
PPDPP$_{\text{user}}$	✓	0.131	10.000	0.918	-
DPDP$_{\text{user}}$	✓	0.269	10.000	0.376	-
TRIP$_{\text{user}}$	✓	0.252	10.000	2.305	-
PADPP	✗	**0.280**	10.000	**2.532**★	-
obj = **Item Frequency**					
DDQN$_{\text{item}}$	✓	0.462	10.000	-	2.501
PPDPP$_{\text{item}}$	✓	0.370	10.000	-	1.828
DPDP$_{\text{item}}$	✓	0.149	10.000	-	1.718
TRIP$_{\text{item}}$	✓	0.412	10.000	-	1.164
PADPP	✗	**0.582**	10.000	-	**2.895**★

Table 3: Empirical results on the **DuRecDial 2.0** dataset. Except for *uniform*, ★ indicates the best performance on the corresponding considered objective.

sentiment due to irrelevant suggestions, while prioritizing user satisfaction alone may miss opportunities to meet the recommendation goal (Lei et al., 2022). In contrast, PADPP adaptively navigates this trade-off. When priorities shift, PADPP again excels on the respective focal metric, achieving the highest *User Sentiment* ($r_{\text{user}} = 2.532^{*}$) and *Item Frequency* ($r_{\text{item}} = 2.895^{*}$) scores. These results further underscore its flexibility and advantage over methods that must retrain for new objectives.

Complexity Analyses. To analyze the computational complexities of PADPP and other baseline methods, we consider a scenario with K distinct objective settings, each allocated a training budget of M training episodes. Obviously, the training complexity of PADPP is $\mathcal{O}(M)$, contrasting with the $\mathcal{O}(KM)$ complexity of baseline approaches. This difference arises since PADPP requires training a single model, whereas existing methods necessitate training K separate models. Additionally, a detailed comparison of training times for PADPP and baseline methods is provided in Appendix A.9.

5.5 Ablation Study

In Figure 3, we show the approximate solution frontiers in two-dimensional objective spaces for PADPP and its variants (PADPP *w/o Know*, PADPP *Min Dist*). Specifically, we approximately evaluate the value functions $\mathbf{V}^{\pi_{\mathbf{w}}}$ of those models across different objective configurations $\mathbf{w}$. First, as our objective is to maximize the objective values, PADPP clearly constructs a better solution frontier as it nearly envelopes the whole solution frontiers established by PADPP *w/o Know*, demonstrating the effectiveness of our knowledge-reuse approach in learning multi-objective dialogue policies. Secondly, we also illustrate the solution frontier established by the *Min Dist* variant. Clearly, the illustrated solution frontiers established by PADPP are generally better than those constructed by these two other variants, demonstrating that GPI can reuse knowledge better than *Min Dist* as we theoretically demonstrated in Section 4.2.2 and Appendix A.1.

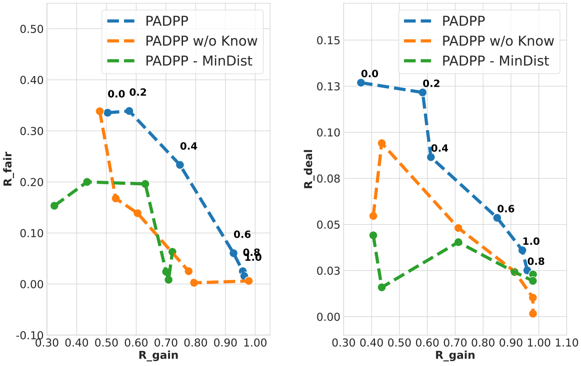


Figure 3: Approximated frontiers established by PADPP, PADPP *w/o Know* and PADPP - *Min Dist*. We show the $r_{\textbf{gain}}, r_{\text{fair}}, r_{\textbf{deal}}$ when varying $\mathbf{w}_{\text{gain}}$.

PADPP	**Deal. A**		**Neg. Equity**		**B. Benefit**	
vs	**Win.(%)**	**Lose.(%)**	**Win.(%)**	**Lose.(%)**	**Win.(%)**	**Lose.(%)**
PPDPP	**38**	24	**53**	28	**32**	24
DPDP	**63**	19	**52**	36	**38**	27
TRIP	**45**	25	**48**	38	**31**	29

Table 4: Human evaluation on Craigslist Bargain dataset. The inter-annotator agreement score is 0.56.

5.6 Human Evaluation

In Table 4, we present the human evaluation results in the uniform weight setting. Overall, our PADPP outperforms all baselines across all evaluated aspects. This improvement is particularly pronounced for **Deal A.** and **Neg. Equity**, suggesting that PADPP-generated conversations prioritize fairness in offers compared to other methods. Conversely, the smaller performance difference observed for **Buyer Benefit** indicates that baseline models place greater emphasis on *Price Gain*.

6 Detailed Analyses

6.1 Handling Trade-offs between Objectives

An adaptive dialogue planner should be capable of handling arbitrary trade-offs between objectives, as determined by the system designer. In Figure 4, we present PADPP's performance under varying weight configurations, each representing a specific trade-off. Specifically, we report the **SR** and reward measures ($r_{\text{gain}}, r_{\text{fair}}, r_{\text{deal}}$) while varying the weights of two objectives and setting the third to zero. The results show that increasing the weight of one objective increases its corresponding objective value while decreasing the values of the other objectives. This demonstrates PADPP's ability to effectively manage arbitrary objective trade-offs. We also examine the impact of objectives on the **SR** metric. SR increases when the weights of either *Fairness* or *Deal Rate* are increased. This is expected, as prioritizing fairness or deal rate increases the likelihood of seller acceptance. Furthermore, we observed an inverse relationship between *Price Gain* and *Deal Rate*: increasing the weight of *Price Gain* decreases r_{deal}, and vice versa. This reflects a typical negotiation dynamic where focusing on individual gain can hinder mutual agreement. This necessitates flexible adjustments in objective priorities, further emphasizing PADPP's adaptability to arbitrary objective trade-offs.

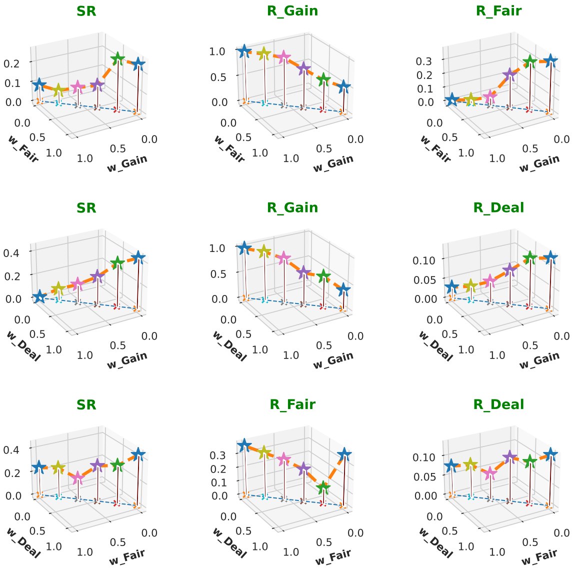


Figure 4: Trade-offs between objectives in negotiation dialogues. We show the SR, $r_{gain}, r_{\text{fair}}, r_{\text{deal}}$ when varying the values of objective weight. The results for recommendation can be found in Appendix A.3.

6.2 Knowledge Reuse Enhances Efficiency

Optimizing MORL learners often necessitates a large number of training examples. Therefore, an effective optimization algorithm needs to leverage training examples effectively. Figure 5 compares PADPP with its ablated variant (PADPP *w/o Know*) under varying amounts of training data, using SR, r_{gain}, r_{fair}, and r_{deal} as metrics. As expected, performance improves for both methods as the training set grows. However, PADPP shows a consistent advantage over PADPP *w/o Know* at all data scales, underscoring the value of knowledge reuse. By distilling insights gleaned from previously trained preference configurations, PADPP achieves faster convergence and higher final scores. In Appendix A.4, we also compare PADPP to baselines (*e.g.*, DDQN, PPDPP, TRIP) at different training sizes, showing that PADPP can match or exceed these baselines when given sufficient data.

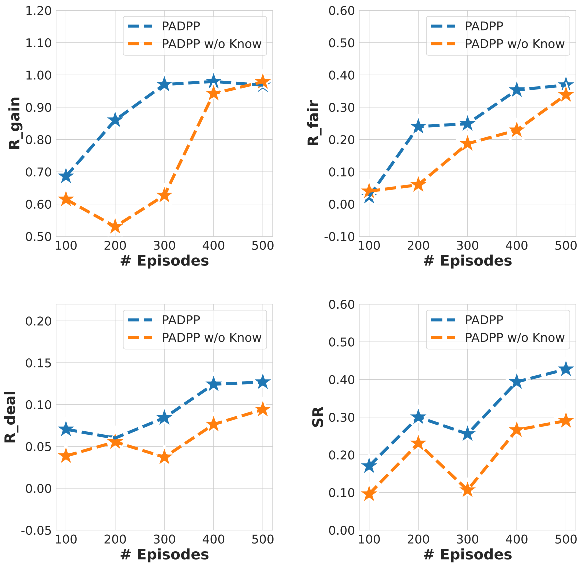


Figure 5: Performance comparison of PADPP and its variant PADPP *w/o Know* across different amounts of training episodes. Specifically, we report the **SR**, $\boldsymbol{r_{\text{gain}}, r_{\text{fair}}, r_{\text{deal}}}$ on **Craigslist Bargain** dataset.

6.3 Impacts of Numbers of Past Preferences

As knowledge reuse is the core of PADPP, it is desired to study the impacts of past update preferences on our proposed method. Specifically, in Figure 6, we report **SR**, r_{gain}, r_{fair}, and r_{deal} within the negotiation scenario across different numbers of past updated preferences ($|\mathcal{W}|$), ranging from 2 to 128. The results for recommendation dialogues can be found in A.7. In particular, the reported results indicate a general performance improvement with increasing $|\mathcal{W}|$ up to a point, after which performance plateaus or slightly declines. Specifically, this plateau or decline is observed beyond $|\mathcal{W}|$ values of 64 or 128. First, this suggests that incorporating more past updated preferences can enhance performance, likely due to the model's increased ability to identify effective *"teachers"* for knowledge reuse. Second, excessively large values of $|\mathcal{W}|$ may introduce noise, negatively impacting per-

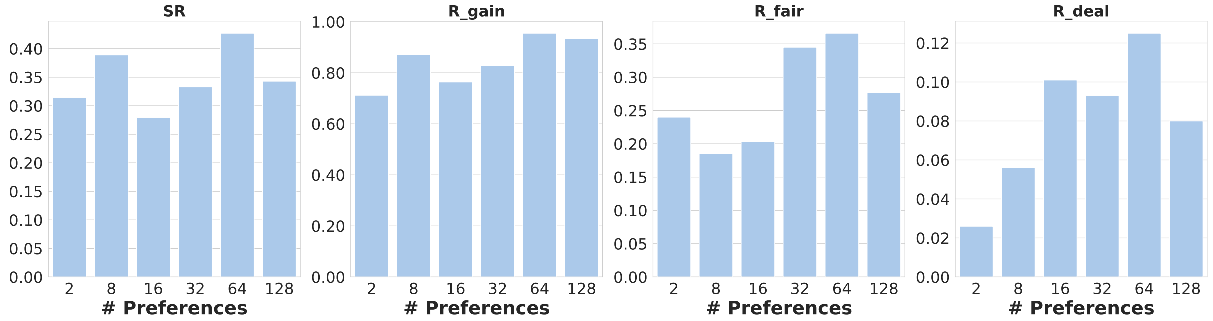
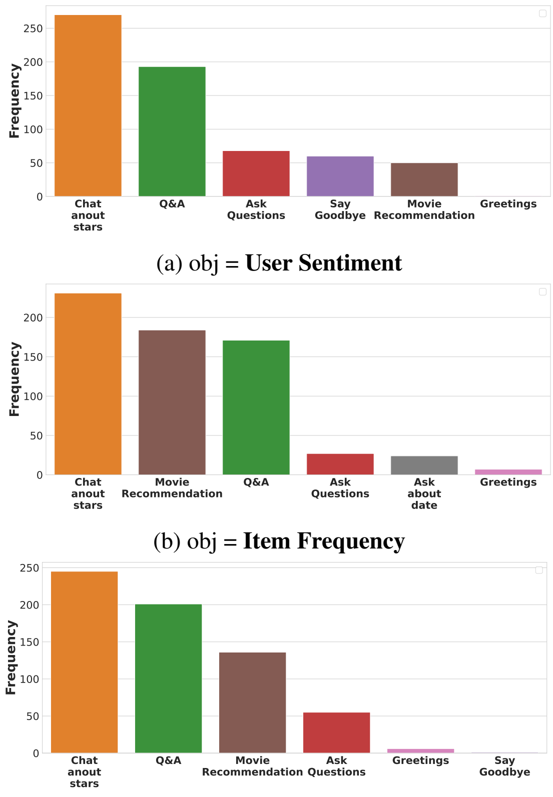


Figure 6: Empirical results of PADPP across different numbers of past updated preferences (i.e., sizes of $\mathcal{W}$). Specifically, we report the results on negotiation dialogues. The results are averaged over 3 different runs.

formance. These results highlight the influence of $|\mathcal{W}|$ on PADPP's effectiveness and the importance of careful selection for optimal results. Notably, all metrics appear to peak at $|\mathcal{W}| = 64$. Therefore, we set $|\mathcal{W}|$ to 64 for all experiments within negotiation dialogues.

6.4 Adaptive Dialogue Strategies



(a) obj = **User Sentiment**

(b) obj = **Item Frequency**

(c) **Uniform**

Figure 7: Most frequent actions employed by PADPP under different objective priorities in recommendation dialogues. We report the statistics in the Movie domain.

To further demonstrate PADPP's capability in adapting its strategies to different objective preferences, in Figure 7, we show the most frequent actions employed by PADPP under different objective priorities, within the recommendation scenario. First, under the obj = **User Sentiment**, PADPP mainly uses actions like *"Chat about stars"*, *"Q&A"*, and *"Ask Questions"*. This indicates that the model prioritizes understanding and clarifying the users' needs to improve their experience. When the priority shifts to maximizing **Item Frequency**, we observe a significant increase in the frequency of *"Movie Recommendation"* action. This is expected, as PADPP adapts its strategy to promote recommendations more directly to users. Finally, in the obj = **Uniform** setting, the model's actions are more balanced. However, *"Chat about stars"* and *"Q&A"* remain common, which indicates a general tendency for PADPP to engage with users and gather information, regardless of the specific objective. These results highlight PADPP's strategic flexibility, which is crucial in real-world situations where objective priorities can change while model retraining is often computationally expensive.

7 Conclusion

In this work, we proposed a novel dialogue policy learning approach called PADPP for multi-objective, goal-oriented dialogues. Specifically, during training, PADPP enhances standard DDQN with a knowledge reuse mechanism, leveraging Generalized Policy Improvement (GPI) for better knowledge transfer. During inference, PADPP could take as its input arbitrary objective configurations and produce corresponding dialogue strategies without retraining the model. Extensive experiments and analyses on two public datasets highlight the superiority and flexibility of PADPP against SOTA dialogue policy approaches.

Limitations

This section explores potential limitations of the proposed PADPP: **(1) Sample Efficiency:** PADPP optimizes the MORL learner across the full range of objective preferences, a process that typically requires extensive training. While the proposed knowledge reuse mechanism improves sample efficiency, substantial training data remain necessary for convergence. **(2) Computational Cost of LLM:** This study leverages LLMs for interactive evaluation and reward computation for MORL, while enhancing ecological validity, introducing a potential limitation. Specifically, the online nature of LLM-based evaluation may result in substantial computational expense. **(3) Multi-objective Reward and Reward Sparsity Problem:** PADPP Our model employs a Double Deep Q-Network (DDQN) as its primary planner, relying heavily on reward signals for optimization. However, in certain domains, obtaining comprehensive multi-objective rewards can be challenging due to reward sparsity. This scarcity of informative reward signals can hinder the learning process.

Acknowledgments

This research is supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative (AISG Award No: AISG-NMLP-2024002). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. 2019. Dynamic weights in multi-objective deep reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 11–20.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650. Association for Computational Linguistics.

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. 2017. Successor features for transfer in reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4058–4068.

Wenqing Chen, Jidong Tian, Caoyun Fan, Yitian Li, Hao He, and Yaohui Jin. 2023. Preference-controlled multi-objective reinforcement learning for conditional text generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, (11):12662–12672.

Huy Dao, Yang Deng, Dung D. Le, and Lizi Liao. 2024a. Broadening the view: Demonstration-augmented prompt learning for conversational recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 785–795.

Huy Dao, Lizi Liao, Dung Le, and Yuxiang Nie. 2023. Reinforced target-driven conversational promotion. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12583–12596.

Huy Quang Dao, Yang Deng, Khanh-Huyen Bui, Dung D. Le, and Lizi Liao. 2024b. Experience as source for anticipation and planning: Experiential policy learning for target-driven recommendation dialogues. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14179–14198.

Yang Deng, Wenqiang Lei, Lizi Liao, and Tat-Seng Chua. 2023a. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. *arXiv preprint arXiv:2305.13626*.

Yang Deng, Wenxuan Zhang, Wai Lam, See-Kiong Ng, and Tat-Seng Chua. 2023b. Plug-and-play policy planner for large language model powered dialogue agents. *Preprint*, arXiv:2311.00262.

Yang Deng, Wenxuan Zhang, Weiwen Xu, Wenqiang Lei, Tat-Seng Chua, and Wai Lam. 2023c. A unified multi-task learning framework for multi-goal conversational recommender systems. *ACM Transactions on Information Systems*, 41(3).

Hado van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, page 2094–2100.

Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*.

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. Association for Computational Linguistics.

Tao He, Lizi Liao, Yixin Cao, Yuanxing Liu, Ming Liu, Zerui Chen, and Bing Qin. 2024a. 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. Association for Computational Linguistics.

Tao He, Lizi Liao, Yixin Cao, Yuanxing Liu, Ming Liu, Zerui Chen, and Bing Qin. 2024b. 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.

Minyoung Hwang, Luca Weihs, Chanwoo Park, Kimin Lee, Aniruddha Kembhavi, and Kiana Ehsani. 2024. Promptable behaviors: Personalizing multi-objective rewards from human preferences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Wenqiang Lei, Yao Zhang, Feifan Song, Hongru Liang, Jiaxin Mao, Jiancheng Lv, Zhenglu Yang, and Tat-Seng Chua. 2022. Interacting with non-cooperative user: A new paradigm for proactive dialogue policy. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 212–222.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Lin Zhao, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. 2023. Summary of ChatGPT-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, and Wanxiang Che. 2021. DuRecDial 2.0: A bilingual parallel corpus for conversational recommendation. In *EMNLP*, pages 4335–4347.

Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020b. Towards conversational recommendation over multi-type dialogs. In *ACL*, pages 1036–1049.

Mary McHugh. 2012. Interrater reliability: The kappa statistic. *Biochemia medica : časopis Hrvatskoga društva medicinskih biokemičara / HDMB*, 22:276–82.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. 2015. Universal value function approximators. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1312–1320.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Hoang V. Dong, Yuan Fang, and Hady W. Lauw. 2025. A contrastive framework with user, item and review alignment for recommendation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, page 117–126.

Aditya Vavre, Ethan He, Dennis Liu, Zijie Yan, June Yang, Nima Tajbakhsh, and Ashwath Aithal. 2024. Llama 3 meets moe: Efficient upcycling. *Preprint*, arXiv:2412.09952.

Jian Wang, Yi Cheng, Dongding Lin, Chak Leong, and Wenjie Li. 2023. Target-oriented proactive dialogue systems with personalization: Problem formulation and dataset curation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1132–1143.

Jian Wang, Dongding Lin, and Wenjie Li. 2022. Follow me: Conversation planning for target-driven recommendation dialogue systems.

Sihan Wang, Kaijie Zhou, Kunfeng Lai, and Jianping Shen. 2020. Task-completion dialogue policy learning via Monte Carlo tree search with dueling network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3461–3471. Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256.

Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. 2019. *A generalized algorithm for multi-objective reinforcement learning and policy adaptation*.

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.

Jun Zhang, Yan Yang, Chencai Chen, Liang He, and Zhou Yu. 2021. KERS: A knowledge-enhanced framework for recommendation dialog systems with multiple subgoals. In *Findings of EMNLP*, pages 1092–1101.

Tong Zhang, Chen Huang, Yang Deng, Hongru Liang, Jia Liu, Zujie Wen, Wenqiang Lei, and Tat-Seng Chua. 2024. 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.

Yangyang Zhao, Zhenyu Wang, Changxi Zhu, and Shihan Wang. 2021. Efficient dialogue complementary policy learning via deep Q-network policy and episodic memory policy. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4311–4323. Association for Computational Linguistics.

A Appendix

A.1 Reusing Knowledge for MORL

Given a randomly sampled objective preference vector $\mathbf{w} \sim \Delta_{|d|}$, our goal is to learn its corresponding state-action value function $\mathbf{Q}(s, a, \mathbf{w}; \theta)$. As discussed in Section 4.2, we could optimize a TD error loss function defined as E.q 1. However, rather than learning $\mathbf{Q}(s, a, \mathbf{w}; \theta)$ from scratch, we aim to leverage knowledge gained from optimizing the model on previously updated preferences $\mathcal{W}$ to expedite the learning process for the new instance $\mathbf{w}$. Therefore, we propose to optimize an additional auxiliary objective function defined as in E.q 2. This approach relies on selecting a teacher policy, denoted as π_{teacher} and an updated preferences $\mathbf{w}_{\text{teacher}} \in \mathcal{W}$ to guide the optimization process for $\mathbf{Q}(s, a, \mathbf{w}; \theta)$. Several methods can be employed to instantiate the teacher policy, which we will discuss later in this Section.

A.1.1 How does GPI-based knowledge reuse work ?

To begin with, we first explain how the knowledge is leveraged using past updated preferences $\mathcal{W}$. Specifically, useful knowledge is encoded by the vector-valued state-action value function $\mathbf{Q}(s, a, \mathbf{w}_i; \theta)$ associated with the policy $\pi_{\mathbf{w}_i}$ corresponding to the past objective preference $\mathbf{w}_i \in \mathcal{W}$. Intuitively, $\mathbf{w}_i^{\mathrm{T}}\mathbf{Q}(s, a, \mathbf{w}_i; \theta)$ quantifies *how good it is to take an action a at the state s if we follow the policy $\pi_{\mathbf{w}_i}$ under preference setting $\mathbf{w}_i$* whereas $\mathbf{w}_j^{\mathrm{T}}\mathbf{Q}(s, a, \mathbf{w}_i; \theta)$ quantifies *how good it is to take an action a at the state s if we follow the policy $\pi_{\mathbf{w}_i}$ under preference setting $\mathbf{w}_j$*. Given the new preference $\mathbf{w}$, we aim to extract an action $\pi_{teacher} \in \mathcal{A}$, suggested by the state-action value functions $\mathbf{Q}(s, a, \mathbf{w}_i; \theta)$ associated with past updated preferences $\mathcal{W}$, whose performance is the highest on the current preference setting $\mathbf{w}$.

Technically, to train DDQN on a preference $\mathbf{w}$, we can employ two consecutive steps, namely **Policy Evaluation (PE)** (i.e, computing $\mathbf{Q}(s, a, \mathbf{w}; \theta)$ for every action $a \in A$) and **Policy Improvement (GI)** (*i.e,* selecting the best action $a \in \mathbf{w}^{\mathrm{T}} \arg\max_{a \in A} Q(s', a, \mathbf{w}; \theta)$) (Eq.1). In this work, we enhance standard PI with **Generalized Policy Improvement (GPI)** (Barreto et al., 2017), which combines multiple different Q functions associated with other learned preferences $\mathcal{W}$ to search for the best action π_{teacher} on the current preference $\mathbf{w}$ (*i.e,* $\pi_{\text{teacher}} \in \arg\max_a \max_{\mathbf{w}_i \in \mathcal{W}} \mathbf{w}^T\mathbf{Q}(s, a, \mathbf{w}_i; \theta)$) (Eq.2). Here, $\mathbf{Q}(s, a, \mathbf{w}_i; \theta)$ is the state-action value function associated with the policy $\pi_{\mathbf{w}_i}$ corresponding to the past objective preference $\mathbf{w}_i \in W$, as explained above. Therefore, we could effectively leverage knowledge gained from training the model on past updated preferences $\mathcal{W}$.

A.1.2 Potential Instantiations of Teacher Policy

Set Max Policy (SMP). Suppose an additional policy network $\pi(s, \mathbf{w}; \theta_\pi)$ (θ_π are the parameters) is learned together with the state-action value function. Then given the preference $\mathbf{w}$, one could instantiate the teacher policy π_{teacher} as the one with the highest performance in the the set $\Pi_{\mathcal{W}} = \{\pi_{\mathbf{w_1}}\}_{i=1}^{|\mathcal{W}|}$. Formally, we could define the set max policy by using the following formulation:

$$\mathbf{w}_{\text{teacher}} = \arg\max_{\mathbf{w}_i \in \mathcal{W}} \mathbf{w}^{\mathrm{T}}\mathbf{Q}(s, \pi(s, \mathbf{w}_i; \theta_\pi), \mathbf{w}_i; \theta^-),$$
$$\pi_{\text{teacher}} \in \pi(s, \mathbf{w}_{\text{teacher}}; \theta_\pi);$$

However, directly learning the policy function $\pi(s, \mathbf{w}; \theta_\pi)$ for MORL is practically challenging and sample inefficient (Hayes et al., 2022). Moreover, we will theoretically demonstrate that our GPI-based knowledge reuse could generally perform no worse than the SMP approach.

Minimum Distance Policy (Min Dist). If a policy $\pi_{\mathbf{w}_i}$ has been established for a preference vector $\mathbf{w}_i$, and the current preference $\mathbf{w}$ is similar to $\mathbf{w}_i$, then $\pi_{\mathbf{w}_i}$ is likely to perform well under $\mathbf{w}$. Therefore, the teacher policy can be selected as the policy optimized for the preference vector closest to $\mathbf{w}$. Formally, we define the minimum distance policy by using the following formulation:

$$\mathbf{w}_{\text{teacher}} = \arg\min_{\mathbf{w}_i \in \mathcal{W}} d(\mathbf{w}_i, \mathbf{w});$$
$$\pi_{\text{teacher}}(s) \in \arg\max_a \mathbf{w}^{\mathrm{T}}\mathbf{Q}(s, a, \mathbf{w}_{\text{teacher}}; \theta^-),$$

where d is a distance function that measures the difference between two objective weight vectors. In practical implementation, we leverage the cosine distance to instantiate the distance function.

Generalized Policy Improvement (GPI). To effectively reuse past knowledge, in this work, we propose to instance the teacher policy by using Generalized Policy Improvement (GPI) (Barreto

Model	Re-Train	SR	Avg. Turn	Avg. Cumulated Rewards r_{gain}	r_{fair}	r_{deal}
Uniform						
Envelope	✗	0.269	9.811	**0.675**	0.181	0.082
PADPP	✗	**0.427**	**9.638**	0.622	**0.287**	**0.142**
obj = **Price Gain**						
Envelope	✗	0.069	9.918	0.835	-	-
PADPP	✗	**0.085**	**9.898**	**0.944**	-	-
obj = **Fairness**						
Envelope	✗	0.226	9.893	-	0.289	-
PADPP	✗	**0.281**	9.792	-	**0.368★**	-
obj = **Deal Rate**						
Envelope	✗	0.186	9.812	-	-	0.077
PADPP	✗	**0.489★**	**9.531**	-	-	**0.165★**

Table 5: Performance comparison between PADPP and Envelope (Yang et al., 2019) on the **Craigslist Bargain** dataset. Except for *uniform*, ★ indicates the best performance on the corresponding considered objective. The final results are reported on 3 different random seeds.

et al., 2017), which is defined as follows:

$$\pi_{\mathbf{w}}^{teacher},$$
$$\mathbf{w}_{\text{teacher}} \in \arg\max_{a,\mathbf{w}_i} \max_{a\in\mathcal{A},\mathbf{w}_i\in\mathcal{W}} \mathbf{w}^{\mathrm{T}}\mathbf{Q}(s,a,\mathbf{w}_i;\theta^{-}),$$

Moreover, leveraging Theorem 1, we demonstrated that the GPI policy is *the best teacher* that we can induce by using the set of state-action value functions $\{\mathbf{Q}^{\pi_{\mathbf{w_i}}}\}_{i=1}^{|\mathcal{W}|}$. Informally, it states that given an objective preference $\mathbf{w}$, the teacher policy deduced by using GPI is no worse than other policies in the set $\Pi_{\mathcal{W}} = \{\pi_{\mathbf{w_1}}\}_{i=1}^{|\mathcal{W}|}$. In other words, the GPI policy should perform no worse than both the SMP and the Min Dist policies.

A.1.3 Connections to Existing MORL Methods

***Remark* 1** *Our PADPP enhances Envelope Update (Yang et al., 2019) by using the set of previously learned solutions.*

Generally speaking, our PADPP can be seen as an extension of (Yang et al., 2019), aiming to maximize the convex envelope of the solution frontier. Specifically, instead of performing GPI over randomly sampled preferences as in (Yang et al., 2019), our method performs envelope updates over the solutions of previously learned ones. Hence, this scheme avoids noisy updates of unlearned weight combinations and makes the learning process converge faster. While Envelope Update (Yang et al., 2019) is not specifically designed for dialogue tasks, we still provide additional experiment results shown in Table 5, comparing PADPP and Envelope Update to verify our hypothesis empirically. Specifically, the results in negotiation dialogues show that PADPP consistently outperforms Envelope Update (Yang et al., 2019) by a considerable margin across different weight settings, demonstrating the superiority of PADPP over the existing MORL algorithm.

A.2 Proofs of Theorem 1

Since $\mathbf{Q}^{\pi_{\mathbf{w}'}}(s,a) \in \mathbb{R}^d$ is the vector-valued action-value function of policy $\pi_{\mathbf{w}'}$ capturing the discounted vector-valued rewards induced by $\pi_{\mathbf{w}'}$ in the environment. Correspondingly, on a new preference weight $\mathbf{w}$, the scalar-valued action-value function $Q_{\mathbf{w}}^{\pi_{\mathbf{w}'}}(s,a) \in \mathbb{R}$ of policy $\pi_{\mathbf{w}'}$ could be computed as:

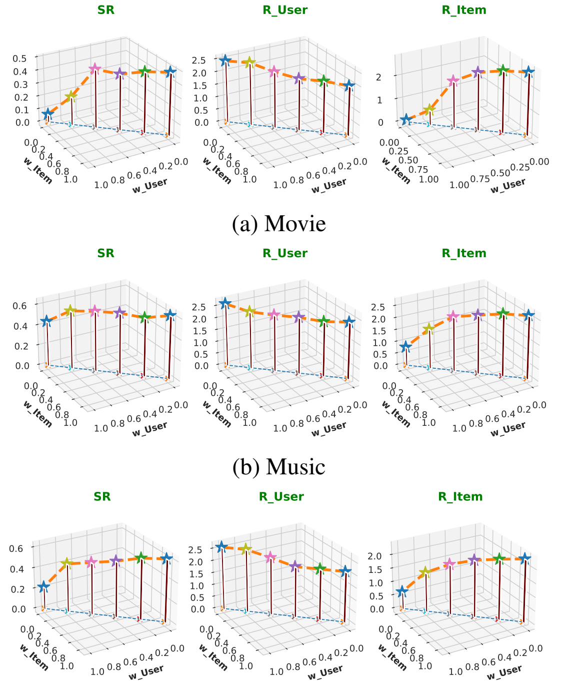
$$\begin{aligned}
&\mathbb{E}_{\pi_{\mathbf{w}'}}\left[\sum_{t=0}^{\infty}\gamma^t\mathbf{w}^{\mathrm{T}}\mathbf{r}(s_t,a_t)|s_0=s,a_0=a\right],\\
&=\mathbf{w}^{\mathrm{T}}\mathbb{E}_{\pi_{\mathbf{w}'}}\left[\sum_{t=0}^{\infty}\gamma^t\mathbf{r}(s_t,a_t)|s_0=s,a_0=a\right],\\
&=\mathbf{w}^{\mathrm{T}}\mathbf{Q}^{\pi_{\mathbf{w}'}}(s,a).
\end{aligned}$$

Intuitively, $Q_{\mathbf{w}}^{\pi_{\mathbf{w}'}}(s,a)$ quantifies how good of taking the action a at the state s if we follow policy $\pi_{w'}$ on the preference configuration $\mathbf{w}$. Assuming we have learned a set of policies $\Pi_{\mathcal{W}} = \{\pi_i\}_{i=1}^{|\mathcal{W}|}$ and their vector-valued action-value functions $\mathbf{Q}^{\pi_i}$ for a set of preferences $\mathcal{W}$. Then given a new preference $\mathbf{w}$, according to the Generalized Policy Improvement (GPI) theorem (Barreto et al., 2017), we have:

$$Q_{\mathbf{w}}^{\pi_{\mathbf{w}}^{gpi}}(s,a) \geq \max_{\mathbf{w}_i\in\mathcal{W}} Q_{\mathbf{w}}^{\pi_{\mathbf{w}_i}}(s,a), \forall s,a \in \mathcal{S}\times\mathcal{A}$$

where $\pi_{\mathbf{w}}^{gpi}(s) \in \arg\max_{a\in\mathcal{A}} \max_{\mathbf{w}_i\in\mathcal{W}} \mathbf{w}^{\mathrm{T}}\mathbf{Q}^{\pi_{\mathbf{w}_i}}(s,a)$. Since $V_{\mathbf{w}}^{\pi}(s) = Q_{\mathbf{w}}^{\pi}(s,\pi(s)) = \mathbf{w}^{\mathrm{T}}\mathbf{Q}^{\pi}(s,\pi(s)) = \mathbf{w}^{T}\mathbf{V}^{\pi}(s)$, we have:

$$\begin{aligned}
Q_{\mathbf{w}}^{\pi_{\mathbf{w}}^{gpi}}(s,a) &\geq \max_{\mathbf{w}_i\in\mathcal{W}} Q_{\mathbf{w}}^{\pi_{\mathbf{w}_i}}(s,a), \forall s,a\\
Q_{\mathbf{w}}^{\pi_{\mathbf{w}}^{gpi}}(s,\pi_{\mathbf{w}}^{gpi}(s)) &\geq \max_{\mathbf{w}_i\in\mathcal{W}} Q_{\mathbf{w}}^{\pi_{\mathbf{w_i}}}(s,\pi_{\mathbf{w}}^{gpi}(s)), \forall s,\\
Q_{\mathbf{w}}^{\pi_{\mathbf{w}}^{gpi}}(s,\pi_{\mathbf{w}}^{gpi}(s)) &\geq \max_{\mathbf{w}_i\in\mathcal{W}} Q_{\mathbf{w}}^{\pi_{\mathbf{w}_i}}(s,\pi_{\mathbf{w}_i}(s)), \forall s,\\
V_{\mathbf{w}}^{\pi_{\mathbf{w}}^{gpi}}(s) &\geq \max_{\mathbf{w}_i\in\mathcal{W}} V_{\mathbf{w}}^{\pi_{\mathbf{w}_i}}(s), \forall s\in\mathcal{S}\\
V_{\mathbf{w}}^{\pi_{\mathbf{w}}^{gpi}}(s) &\geq V_{\mathbf{w}}^{\pi_{\mathbf{w}_i}}(s), \forall s\in\mathcal{S}, \mathbf{w}_i\in\mathcal{W},\\
\mathbb{E}_{s_0\sim\mu}[V_{\mathbf{w}}^{\pi_{\mathbf{w}}^{gpi}}(s_0)] &\geq \mathbb{E}_{s_0\sim\mu}[V_{\mathbf{w}}^{\pi_{\mathbf{w}_i}}(s_0)], \forall \mathbf{w}_i\in\mathcal{W},\\
V_{\mathbf{w}}^{\pi_{\mathbf{w}}^{gpi}} &\geq V_{\mathbf{w}}^{\pi_{\mathbf{w}_i}}, \forall \mathbf{w}_i\in\mathcal{W},\\
\mathbf{w}^{\mathrm{T}}\mathbf{V}^{\pi_{\mathbf{w}}^{gpi}} &\geq \mathbf{w}^{\mathrm{T}}\mathbf{V}^{\pi_{\mathbf{w}_i}}, \forall \mathbf{w}_i\in\mathcal{W},
\end{aligned}$$



(a) Movie

(b) Music

(c) POI

Figure 8: Trade-off between objectives in recommendation dialogues. Specifically, we show the SR, $r_{\textbf{user}}$, r_{user} when varying the values of the objective weight. The final results are reported on 3 different random seeds.

Hence, under the preference $\mathbf{w}$, the GPI policy $\pi^{gpi}_{\mathbf{w}}(s)$ dominates other policies in the set $\Pi_{\mathcal{W}}$.

A.3 Handling Trade-offs between Objectives: Recommendation Dialogues

Figure 8 depicts the trade-off between multiple objectives within recommendation dialogues across three distinct domains: Movie (a), Music (b), and Points of Interest (POI) (c). Each subfigure presents a three-dimensional plot where the x and y axes correspond to the weights assigned to *Item Frequency* ($\mathbf{w}_{\text{item}}$) and *User Sentiment* ($\mathbf{w}_{\text{user}}$) objectives, respectively. The z-axis represents the performance of our PADPP with respect to three key metrics: Success Rate (**SR**) and the objective values ($r_{\text{user}}, r_{\text{item}}$). Across all domains, a consistent trend is observed: increasing the weight assigned to a specific objective results in improved performance on that objective, while concurrently diminishing performance on the other objective. These results provide further evidence of the capability of PADPP to effectively manage trade-offs between multiple objectives within recommendation scenarios.

Across all domains, experimental results indicate that increasing the weight of item frequency generally leads to an increase in success rate (SR) and a slight decrease in user return (r_{user}). This observation is reasonable, as a higher recommendation tendency is likely to improve the chances of achieving the recommendation target, while simultaneously increasing the likelihood of users rejecting some inappropriate recommendations. This raises a dilemma between user satisfaction and goal completion as indicated in (Lei et al., 2022), necessitating careful consideration of the trade-offs between these two aspects. Furthermore, in practical applications, these trade-offs frequently change based on varying circumstances and end users, requiring dialogue planners to adapt to dynamic objective preferences. This further highlights the impact of PADPP's capability in real-world recommendation scenarios.

A.4 Performance Comparison Across Different Numbers of Training Episodes

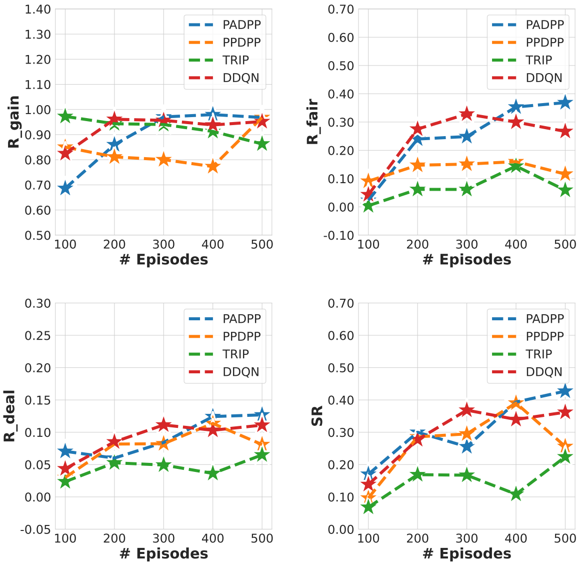


Figure 9: Performance comparison of PADPP and other baseline methods across different amounts of training episodes. Specifically, we report the **SR**, $r_{\textbf{gain}}$, $r_{\textbf{fair}}$, $r_{\textbf{deal}}$ on **Craigslist Bargain** dataset.

In Figure 9, we present the performance comparison between the proposed PADPP method and three baseline methods (PPDPP, TRIP, and DDQN) on the Craigslist Bargain dataset. Performance is evaluated using four metrics: **SR**, r_{gain}, r_{fair}, and r_{deal}. Each subplot in the figure depicts the trend of these metrics as a function of training episodes, ranging from 100 to 500. Initially, the baseline methods outperform PADPP with fewer training episodes. This may be attributed to the fact that the baseline methods optimize for a single objective configuration, whereas PADPP learns policies across multiple objective settings concurrently. This observation underscores the challenge of sample efficiency in Multi-Objective Reinforcement Learning (MORL). As discussed in Section 6.2, we show that our knowledge-reusing mechanism enhances sample efficiency in MORL training. Furthermore, the presented results also demonstrate that PADPP achieves competitive or superior performance compared to the baselines, especially with increasing training episodes. This result aligns with our hypothesis that learning multi-objective policies across the whole spectrum of objective preferences requires a substantial number of training examples.

A.5 Performance Comparison Across Different Recommendation Domains

In Table 6, we present a performance comparison of the proposed PADPP method and baseline methods across Movie, Music, and Point-of-Interest (POI) recommendation domains. Initially, we investigate the performance on the **Uniform** setting, where objective importance is distributed equally. Consistent with Section 5.4, PADPP effectively balances the two objectives, namely *User Sentiment* and *Item Frequency*, across all domains, while baseline methods tend to favor individual objectives. Additionally, although PADPP achieves the best performance on only two metrics $\mathbf{SR}$ and r_{item} in Movie and POI domains, respectively, it still shows second-best results on 7 out-of-9 metrics (excluding Average Turn ($\mathbf{Avg.Turn}$)). This demonstrates PADPP's superior planning capabilities compared to state-of-the-art approaches.

Subsequently, we assess the model performance under shifting objective priorities (i.e. *Uniform* $\rightarrow$ *User Sentiment, Item Frequency*). Specifically, when *User Sentiment* was prioritized, PADPP achieves the highest results on the corresponding objective across all domains ($r_{\text{user}} = \mathbf{2.663}^{*}, \mathbf{2.475}^{*}, \mathbf{2.489}^{*}$ for Movie, Music, and POI, respectively. With Item Frequency as the priority, PADPP yields the best performance on the corresponding objective in the Movie domain ($r_{\text{item}} = \mathbf{3.761}^{*}$) and competitive performance with PPDPP and DDQN in the Music and POI domains, respectively. This further highlights PADPP's adaptability to changing objective priorities. Finally, unlike other baselines requiring retraining for adjusted objective priorities, PADPP adapted directly without retraining, demonstrating its computational advantages.

A.6 Performance of Comparison Across Different Numbers of Conversation Turns

In Figure 10, we present the average cumulative rewards for PADPP and baseline methods across conversation turns. Specifically, we report $r_{\text{gain}}, r_{\text{fair}}, and r_{\text{deal}}$ under uniform weight settings. All models exhibit an initial increase in grain followed by a subsequent decrease. This trend suggests a common negotiation tactic: buyers may initially propose low prices, increasing their offers in later rounds to facilitate agreement. The concurrent increase in r_{fair} and r_{deal} supports this interpretation. PADPP shows strong performance across all three metrics and throughout the negotiation. Conversely, the baseline models exhibit a tendency to prioritize individual objectives. For instance, TRIP and DDQN primarily optimize r_{gain} at the expense of $r_{\text{fair}} and r_{\text{deal}}$. PPDPP, conversely, prioritizes r_{fair} and r_{deal}, resulting in lower r_{gain}. Our proposed model, however, achieves the best balance among these objectives. These findings demonstrate our model's effectiveness in managing multiple, often competing, objectives.

A.7 Performance of PADPP Across Different Sizes of Past Updated Preferences

In this section, we investigate the impacts of the number of past updated preferences on the performance of PADPP within the recommendation scenarios. In Figure 11, we present the performance of PADPP across different numbers of past updated preferences, denoted by $|\mathcal{W}|$, from 2 to 128. Specifically, we report $\mathbf{SR}$, r_{user}, and r_{item} across three recommendation domains: Movie, Music, and POI. In particular, we observe that performance across all metrics generally increases with larger $|\mathcal{W}|$, indicating that incorporating more past preferences might enhance knowledge reuse. However, excessively large $|\mathcal{W}|$ values degrade performance, likely due to increased noise in the optimization process. Finally, we observe that the best performance is achieved at $\mid \mathcal{W} \mid$ = 32. Consequently, we set $|\mathcal{W}|$ to 32 for all recommendation experiments.

A.8 Detailed Descriptions regarding Multi-objective Dialogue Environments

This section details the dialogue environments employed in this study. Specifically, we investigate two multi-objective dialogue scenarios: negotiation (He et al., 2018) and recommendation (Liu et al., 2021; Dao et al., 2024a; V. Dong et al., 2025). Illus-

Model		Movie				Music				POI			
				Avg. Cumulated Rewards				Avg. Cumulated Rewards				Avg. Cumulated Rewards	
	Re-Train	SR	Avg. Turn	$\mathbf{r}_{\text{user}}$	r_{item}	SR	Avg. Turn	$\mathbf{r}_{\text{user}}$	r_{item}	SR	Avg. Turn	$\mathbf{r}_{\text{user}}$	r_{item}
Uniform													
DDQN_{uni}	✓	0.070	10.000	1.193	0.427	0.230	10.000	1.266	1.037	**0.564**	10.000	0.590	**2.351**
$\text{PPDPP}_{\text{uni}}$	✓	0.244	10.000	-0.070	1.612	0.319	10.000	0.920	1.848	0.179	10.000	0.595	0.637
DPDP_{uni}	✓	0.188	10.000	-0.427	**3.142**	0.405	10.000	0.957	**2.364**	0.000	10.000	0.989	0.266
TRIP_{uni}	✓	0.546	10.000	**2.229**	0.825	0.186	10.000	2.163	1.142	0.000	10.000	**3.669**	0.000
PADPP (Ours)	✗	0.438	10.000	1.872	2.827	**0.676**	10.000	**2.702**	2.022	0.403	10.000	2.123	1.768
obj = **User Sentiment**													
$\text{DDQN}_{\text{user}}$	✓	0.005	10.000	1.376	-	0.113	10.000	1.228	-	0.025	10.000	0.958	-
$\text{PPDPP}_{\text{user}}$	✓	0.064	10.000	0.793	-	0.295	10.000	0.949	-	0.034	10.000	1.012	-
$\text{DPDP}_{\text{user}}$	✓	0.265	10.000	-0.436	-	0.259	10.000	0.999	-	0.283	10.000	0.565	-
$\text{TRIP}_{\text{user}}$	✓	**0.351**	10.000	2.283	-	0.171	10.000	2.305	-	0.234	10.000	2.327	-
PADPP (Ours)	✗	0.004	10.000	**2.663**★	-	**0.302**	10.000	**2.475**★	-	**0.533**	10.000	**2.459**★	-
obj = **Item Frequency**													
$\text{DDQN}_{\text{item}}$	✓	0.207	10.000	-	2.480	0.402	10.000	-	1.992	**0.796**	10.000	-	**3.032**★
$\text{PPDPP}_{\text{item}}$	✓	0.269	10.000	-	1.916	0.453	10.000	-	**2.774**★	0.390	10.000	-	1.514
$\text{DPDP}_{\text{item}}$	✓	0.231	10.000	-	3.074	0.216	10.000	-	1.280	0.000	10.000	-	0.800
$\text{TRIP}_{\text{item}}$	✓	0.515	10.000	-	0.853	0.504	10.000	-	2.198	0.219	10.000	-	1.458
PADPP (Ours)	✗	**0.554**	10.000	-	**3.761**★	**0.658**	10.000	-	2.710	0.533	10.000	-	2.342

Table 6: Empirical results on the **DuRecDial 2.0** dataset. We report the performance comparison on three domains, namely **Movie**, **Music**, and **POI** recommendation. Except for *uniform*, ★ indicates the best performance on the corresponding considered objective. The final results are reported on 3 different random seeds.

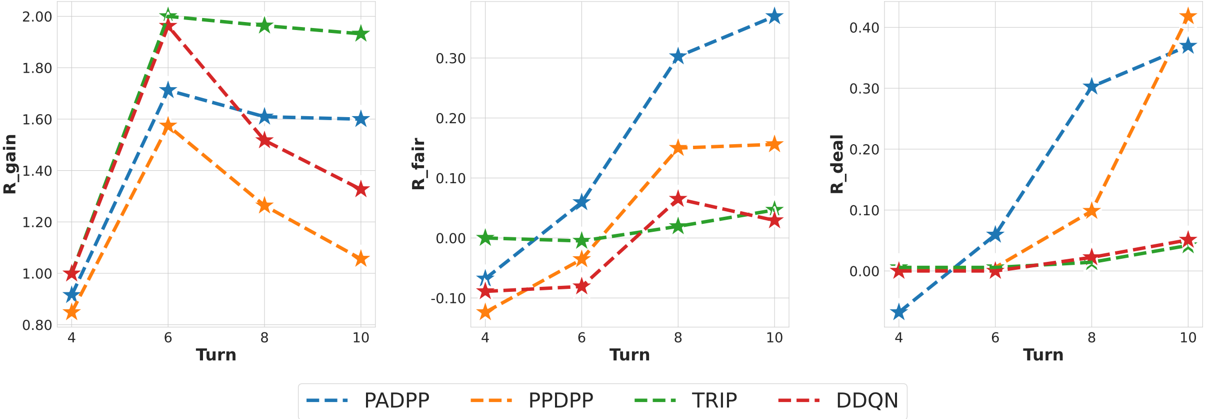


Figure 10: Avg. cumulated rewards at different conversation turns of PADPP and other baseline methods in the **Uniform** setting. Specifically, we report the $r_{\text{gain}}, r_{\text{fair}}, r_{\text{deal}}$ on the **Craigslist Bargain**.

trative examples of these scenarios are presented in Figures 12 and 13, respectively. Unlike prior work (Deng et al., 2023b; He et al., 2024a; Zhang et al., 2024), which addresses single or fixed objective combinations, this work tackles the multi-objective dialogue planning problem under dynamic objective preferences, a more complex and realistic challenge. The subsequent sections describe the multi-objective reward computation protocols for each dialogue scenario.

A.8.1 Multi-objective Negotiation Dialogue

For negotiation, a buyer and a seller converse in a product bargaining situation to reach a common agreement, as shown in Figure 12. For this scenario, we consider 3 distinct objectives, including *Price Gain*, *Fairness*, and *Deal Rate*, where *Price Gain*, *Fairness* are considered as two conflicting objectives. Formally, the reward computations for those objectives are defined as follows:

- r_{gain}: At each turn, if the buyer (i.e, our dialogue agent) either proposes a new price or counters the user with a counter price, we utilize a simple regular expression to extract the mentioned price and compute r_{gain} as follows:

$$r_{\text{gain}} = \frac{p - p_{seller}}{p_{buyer} - p_{seller}} \tag{4}$$

where r_{gain} is greater if the mentioned price is closer to the buyer's initial price. If no price appears in the system response, then $r_{\text{gain}} = 0$. Intuitively, a higher value of r_{gain} means a greater advantage for the buyer.

- r_{fair}: We regard the middle point of the buyer and seller's initial prices as a fair offer. Hence, at each turn, if the buyer provides a price, then we extract the mentioned price and compute the r_{fair} as follows:

$$r_{\text{fair}} = \alpha_{\text{fair}} - \frac{\mid p - p_{mid} \mid}{p_{seller} - p_{buyer}}, \tag{5}$$

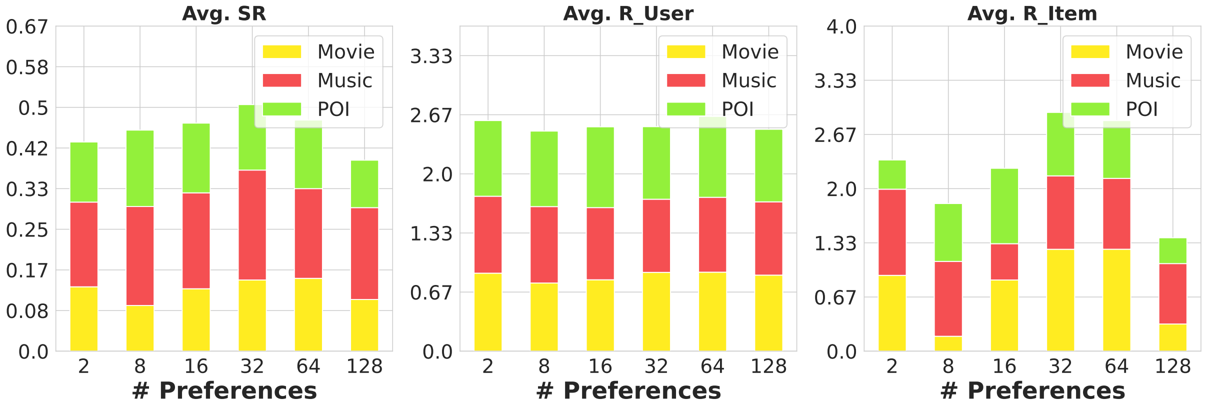
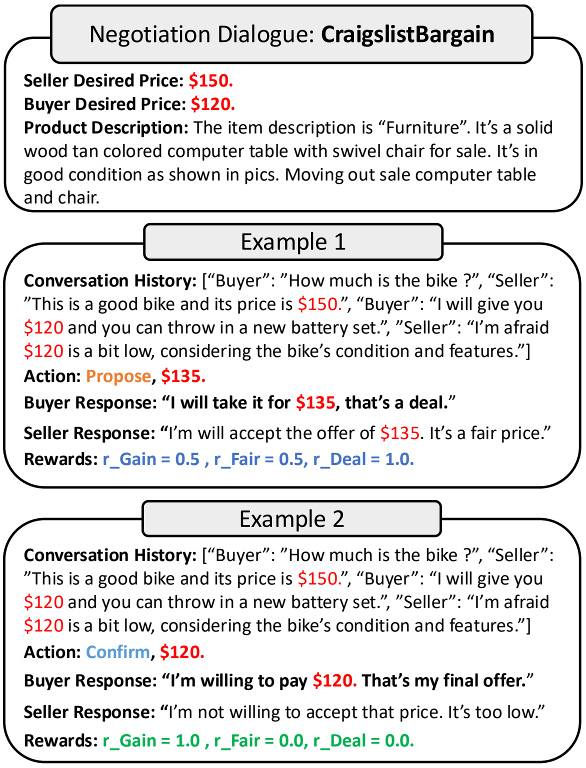


Figure 11: Empirical results of PADPP across different numbers of past updated preferences (i.e., sizes of $\mathcal{W}$). Specifically, we report the averaged results on 3 different runs.



Negotiation Dialogue: **CraigslistBargain**

Seller Desired Price: $150.

Buyer Desired Price: $120.

Product Description: The item description is "Furniture". It's a solid wood tan colored computer table with swivel chair for sale. It's in good condition as shown in pics. Moving out sale computer table and chair.

Example 1

Conversation History: ["Buyer": "How much is the bike ?", "Seller": "This is a good bike and its price is $150.", "Buyer": "I will give you $120 and you can throw in a new battery set.", "Seller": "I'm afraid $120 is a bit low, considering the bike's condition and features."]

Action: **Propose**, **$135.**

Buyer Response: "I will take it for $135, that's a deal."

Seller Response: "I'm will accept the offer of $135. It's a fair price."

Rewards: r_Gain = 0.5 , r_Fair = 0.5, r_Deal = 1.0.

Example 2

Conversation History: ["Buyer": "How much is the bike ?", "Seller": "This is a good bike and its price is $150.", "Buyer": "I will give you $120 and you can throw in a new battery set.", "Seller": "I'm afraid $120 is a bit low, considering the bike's condition and features."]

Action: Confirm, **$120.**

Buyer Response: "I'm willing to pay $120. That's my final offer."

Seller Response: "I'm not willing to accept that price. It's too low."

Rewards: r_Gain = 1.0 , r_Fair = 0.0, r_Deal = 0.0.

Figure 12: An example of a multi-objective negotiation dialogue. The reward computation for negotiation is described in Section A.8.1.

where $p_{mid} = \frac{p_{seller}+p_{buyer}}{2}$ and $\alpha_{\text{fair}} > 0$ is a predefined hyper-parameter ($\alpha_{\text{fair}} = 0.5$ for the default configuration). Intuitively, the fairness reward is greater if the mentioned price is closer to the middle price and vice versa.

- r_{deal}: Following existing works (Deng et al., 2023b; He et al., 2024a), at each turn, we assess if the seller and the buyer reach a deal or not. Specifically, we prompt an LLM (Liu et al., 2023) for N times and convert the textual outputs to scalar values. Formally, the deal reward r_{deal} is computed as follows:

$$r_{\text{deal}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{V}(\text{LLM}(\mathcal{P}_{\text{deal}}, \mathcal{H})) \qquad (6)$$

where $\mathcal{V}$ is a function that converts a textual output to a scalar value. $\mathcal{H}$ is the current conversation history and $\mathcal{P}_{\text{deal}}$ is designated prompt described as in Figure 18. We regard a conversation as a successful one if the $r_{\text{deal}} \geq \epsilon_{\text{deal}}$.

A.8.2 Multi-objective Recommendation Dialogue

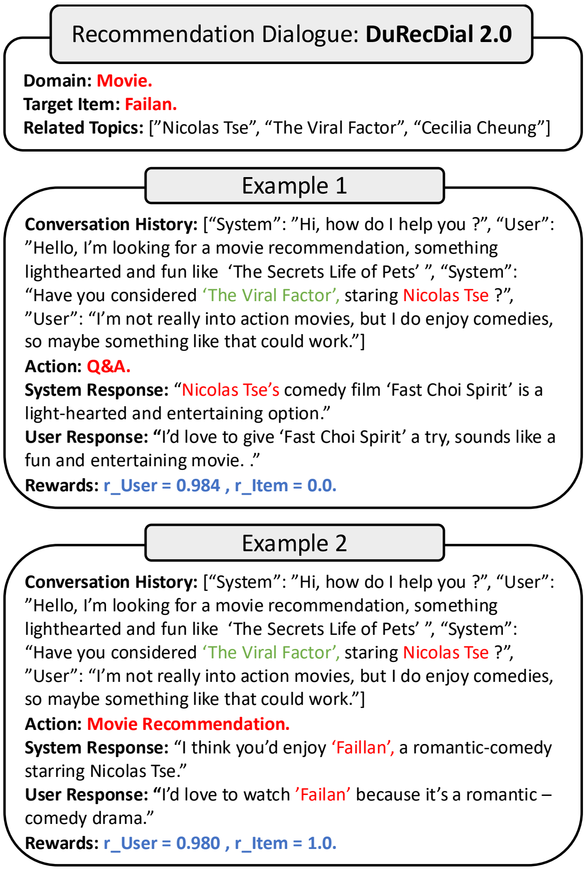
For the recommendation scenario, each conversation is associated with an item v. Given a specific item v and a set of related background knowledge $\mathcal{K}$, our goal is to recommend the item v to the user, as shown in Figure 13. More particularly, for this scenario, we consider two distinct objectives, including *User Sentiment* (r_{user}) and *Item Frequency* (r_{item}), which are defined as follows:

- r_{user}: At each turn, after the system takes action, given the user's generated response y_{user}, we utilize a 3-classes pre-trained RoBERTa model for sentiment analysis (Barbieri et al., 2020) [2] to produce a user sentiment score. Formally, the user sentiment reward r_{user} is computed as follows:

$$s_{\text{senti}}, c_{\text{senti}} = \text{RoBERTa}_{\text{sentiment}}(y_{\text{user}}),$$

$$r_{\text{user}} = \begin{cases} s_{senti}, & \text{if } c_{\text{senti}} = \text{"Positive"}, \\ 0, & \text{if } c_{\text{senti}} = \text{"Neutral"}, \\ -s_{senti}, & \text{if } c_{\text{senti}} = \text{"Negative"}, \end{cases}$$

[2] https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment



Recommendation Dialogue: **DuRecDial 2.0**

Domain: **Movie.**
Target Item: **Failan.**
Related Topics: ["Nicolas Tse", "The Viral Factor", "Cecilia Cheung"]

Example 1

Conversation History: ["System": "Hi, how do I help you ?", "User": "Hello, I'm looking for a movie recommendation, something lighthearted and fun like 'The Secrets Life of Pets' ", "System": "Have you considered 'The Viral Factor', staring Nicolas Tse ?", "User": "I'm not really into action movies, but I do enjoy comedies, so maybe something like that could work."]
Action: **Q&A.**
System Response: "Nicolas Tse's comedy film 'Fast Choi Spirit' is a light-hearted and entertaining option."
User Response: "I'd love to give 'Fast Choi Spirit' a try, sounds like a fun and entertaining movie. ."
Rewards: **r_User = 0.984 , r_Item = 0.0.**

Example 2

Conversation History: ["System": "Hi, how do I help you ?", "User": "Hello, I'm looking for a movie recommendation, something lighthearted and fun like 'The Secrets Life of Pets' ", "System": "Have you considered 'The Viral Factor', staring Nicolas Tse ?", "User": "I'm not really into action movies, but I do enjoy comedies, so maybe something like that could work."]
Action: **Movie Recommendation.**
System Response: "I think you'd enjoy 'Faillan', a romantic-comedy starring Nicolas Tse."
User Response: "I'd love to watch 'Failan' because it's a romantic – comedy drama."
Rewards: **r_User = 0.980 , r_Item = 1.0.**

Figure 13: An example of a multi-objective recommendation dialogue. The reward computation for negotiation is described in Section A.8.2.

where s_{senti}, c_{senti} are the predicted sentiment score and class, respectively.

- r_{rec}: At each turn, given the predicted action a and the system's generated response y_{sys} by the system, the reward for the second objective *Item Frequency* r_{item} is computed as follows:

$$r_{\text{item}} = \begin{cases} \beta, & a \in \mathcal{A}_{rec}, \\ \beta + \delta, & a \in \mathcal{A}_{rec}, v \in y_{\text{sys}}, \end{cases}$$

where $\beta, \delta > 0$ are predefined hyperparameters, δ is an extra reward obtained if the target item v is recommended to the user. $\mathcal{A}_{rec} \in \mathcal{A}$ is the set of recommendation-centric actions, which is defined as in Table 12.

In recommendation scenarios, we evaluate if a conversation is a successful one by following two criteria. First, the target item v must be present within the system's response. Second, similar to negotiation dialogue evaluation, an LLM is queried by N times to determine user acceptance of item v. The LLM's textual outputs are then converted into scalar values. Subsequently, a conversation is considered successful if the average of these scalar values exceeds a predefined threshold, ϵ_{rec}. Additionally, the specific prompting strategy for this evaluation is detailed in Figure 19.

A.9 Computational Complexities

In this study, we primarily investigate the problem of learning dialogue policies across dynamic objective weight configurations. Specifically, dialogue policies are learned for K distinct objective settings ($K = 4$ for negotiation and $K = 3$ for recommendation), with M training episodes per setting ($M = 500$ for the default configuration). In table 7, we first present the computational complexity (in terms of Big-O notation) and total training time for the proposed method and baseline approaches. Specifically, the proposed method exhibits a computational complexity of $\mathcal{O}(M)$, as it is trained once and adapts to varying objective weights during inference. In contrast, the other baselines require training for each objective configuration, resulting in $\mathcal{O}(KM)$ complexity. Critically, these baselines require retraining when new objective settings are introduced, highlighting their computational inefficiency and the importance of adaptability in dialogue policy learning.

Empirical computation time (including training and evaluation) in Table 7 supports these analyses. The proposed method achieves the lowest computational time, consistent with its single-training paradigm, while the baselines require retraining for each objective weight modification.

Model	Complexity	Total Computation Time (hours)
DDQN	$\mathcal{O}(KM)$	9.91
PPDPP	$\mathcal{O}(KM)$	9.92
TRIP	$\mathcal{O}(KM)$	15.31
PADPP (ours)	$\mathcal{O}(M)$	**3.71**

Table 7: Computational complexity and total computation time ($K = 4, M = 500$) of PADPP and other baseline methods. In particular, the maximal number of conversation turns for **Craigslist Bargain** is set to 10.

A.10 Implementation Details

In this work, we implement our proposed PADPP using the PyTorch framework [3]. All experiments were conducted on 8 NVIDIA L40 40GB GPU cards. Moreover, the performance evaluation protocols are detailed as follows:

- **Negotiation Dialogue**: In this study, we adopt the original data split from He et al. (2018). The training split was used to train our model.

[3] https://pytorch.org/

During training, objective configurations were randomly sampled from a d-dimensional probability simplex, denoted as Δ_d. Model selection was performed by choosing the checkpoint that yielded the best performance on the validation set under a uniform weight setting. The final evaluation was conducted on the test set using the selected checkpoint.

- **Recommendation Dialogue**: The DuRecDial 2.0 dataset (Liu et al., 2021) encompasses conversations across several domains, namely Movie, Music, and POI Recommendation. Consequently, we first partition the dataset based on these domains. The models are trained and evaluated on each resulting subset to determine domain-specific performance. These individual performance metrics were then averaged to yield an overall performance score. The training, validation, and testing procedures followed those established for the negotiation scenario.

For PADPP's model configurations, for a fair comparison, we follow previous works (Deng et al., 2023b; He et al., 2024a) utilizing RoBERTa-Large (Liu et al., 2020a) [4] (343M) as the backbone for the policy planner. Then we add a shallow Multilayer Perceptron (MLP) on top of the pre-trained language model to instantiate the Q network. Formally, given a state s, we first pass the state through RoBERTa to obtain a hidden state $h \in \mathbb{R}^{1024}$. Afterward, we project the hidden state h to a low-dimensional feature vector $\hat{h} \in \mathbb{R}^{128}$. Then we project the feature vectors $\hat{h}$ to an output tensor of size $|\mathcal{A}| \times d$ ($|\mathcal{A}|$ is the size of the action space, d is the number of objectives), representing multiple-objective Q values. Moreover, following existing methods (Deng et al., 2023b), we first pre-train the Q-network with supervised learning on the corresponding background dataset (i.e,. DuRecDial 2.0 or Craigslist Bargain) to predict the next action. For pertaining, we set the number of training epochs to 10 and 2 for DuRecDial 2.0 and Craigslist Bargain, respectively. Moreover, for both datasets, we utilize a training batch size of 8 and fine-tune the model with a learning rate of 5e-5. After supervised pre-training, we fine-tune the Q network with MORL as described in Section 4.2. Specifically, we utilize a learning rate of 5e-4 and fine-tune the model for $M = 500$ training episodes. For the LLM component utilized in this work, we leverage Llama-3 (8B) (Vavre et al., 2024) [5], an open-source LLM, for user simulator, response generation, and reward computation. The detailed hyperparameters of PADPP can be found in Table 8.

Hyper-parameters	**DuRecDial 2.0**	**Craislist Bargain**
Phase: Supervised Pretraining		
# batch size	8	8
max tokens length	512	512
learning rate	5e-5	5e-5
# epochs	10	2
dropout	0.1	0.1
Phase: MORL Finetuning		
# objectives	2	3
# episodes M	500	500
# batch size	128	128
learning rate	5e-4	5e-4
# updated preferences $\|\mathcal{W}\|$	32	64
balancing param α	0.7	0.7
ϵ_{deal}	-	1.0
ϵ_{rec}	1.0	-
N	5	10
# Turns	10	10
discount factor γ	0.99	0.99
buffer size	2000	2000

Table 8: Detailed implementation of PADPP on the Craigslist Bargain and DuRecDial 2.0 datasets.

Regarding objective configurations for performance evaluation, for the negotiation scenario, the objective weights are $\mathbf{w}_{gain} = [1, 0, 0], \mathbf{w}_{fair} = [0, 1, 0], \mathbf{w}_{deal} = [0, 0, 1]$ for *price gain*, *fairness*, and *deal rate*, respectively. For recommendation, the weights are $\mathbf{w}_{user} = [1, 0], \mathbf{w}_{item} = [0, 1]$ for *user sentiment* and *item frequency*, respectively. Finally, for the uniform setting, the weights are $\mathbf{w}_{uni} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ for negotiation and $\mathbf{w}_{uni} = [\frac{1}{2}, \frac{1}{2}]$ for recommendation.

A.11 Additional Information regarding Baseline Methods

In this section, we provide additional details regarding baseline methods. Specifically, in this work, we compare our proposed method PADPP against various RL baseline approaches, including:

- **DDQN** (Hasselt et al., 2016) is the standard Double Deep Q Network approach.
- **PPDPP** [6] (Deng et al., 2023c) is a recent LLM dialogue agent. In particular, this model leverages some background datasets to fine-tune a small LM model, serving as the prior dialogue policy. The policy is then further fine-tuned with simulated conversations generated via RL to maximize long-term rewards.

[4] https://huggingface.co/FacebookAI/roberta-large

[5] https://huggingface.co/meta-llama/Meta-Llama-3-8B

[6] https://github.com/dengyang17/PPDPP

- **DPDP** [7] (He et al., 2024a) is another LLm-based dialogue agent. In particular, it first employs a pre-training step in which the dialogue policy planner is pre-trained using offline reinforcement learning. Afterward, it enhances the policy planner with a self-play fine-tuning method using Monte-Carlo Tree Search (MCTS).
- **TRIP** (Zhang et al., 2024) is a more recent plug-in policy planner, which enhances dialogue strategy learning with user-centric mental information. Moreover, it utilizes various user simulators to fine-tune the planner with RL.

For the SOTA plug-in policy planners, we leverage their published source codes to conduct experiments. In contrast to our PADPP, in those baseline methods, the desired preference vector $\mathbf{w}$ (*i.e.* it is identical to $\mathbf{w}_{\text{infer}}$ in our method) needs to be pre-defined at training time so that the scalarized reward signal can be computed accordingly. Consequently, such single-objective methods necessitate retraining upon changing objective configurations. Finally, for the ablation study, we compared our PADPP with other variants, described as follows:

- **PADPP** *W/o Know*: The variant without the knowledge reuse mechanism.
- **PADPP** - *Min Dist*: The variant utilizing **Minimum Distance Policy** approach for knowledge reuse.

A.12 Additional Statistics of Benchmark Datasets

In this work, we adopted two published goal-oriented datasets, namely Craigslist Bargain (He et al., 2018) and DuRecDial 2.0 (Liu et al., 2021). In fact, these two benchmarks have been widely utilized to evaluate recent plug-in policy planners, such as PPDPP (Deng et al., 2023b), DPDP (He et al., 2024b), and TRIP (Zhang et al., 2024). Moreover, those two datasets comprise dialogues centered on two practical scenarios, namely price negotiation and item recommendation, often involving multiple, often competing, objectives. Such a characteristic makes them appropriate for evaluating multi-objective dialogue policy methodologies.

Additionally, in Table 9, we present additional statistics for the DuRecDial 2.0 dataset across the Movie, Music, and POI domains. Specifically, we report the number of actions and the distribution of dialogue cases across the training, development, and test sets. The reported statistics reveal that the Music domain contains the most actions (11), followed by Movie (8) and POI (5). The number of dialogue cases also varies by domain, with Movie having the most and POI the fewest, indicating potential data scarcity for the POI domain. This data imbalance suggests that learning effective dialogue policies may be more difficult for POI recommendations. This hypothesis is supported by the results in Section A.5, which show the lowest average Success Rate (**SR**) for POI recommendations under the uniform weight setting.

Additionally, in Table 10, we present the distribution of dialogue strategies within the DuRecDial 2.0 and Craigslist Bargain datasets. First, for DuRecDial 2.0, consistent with the statistics shown in Table 9, the Movie and Music domains contain the most data (# Music Recommendation = 13,170, # Movie Recommendation = 14,807). In contrast, the POI domain (# POI Recommendation = 5,448) may present a data scarcity issue, potentially increasing the difficulty of model training. Secondly, regarding the Craigslist Bargain dataset, "active" actions (e.g., "counter," "propose," and "inquire") are prevalent, suggesting extensive communication between sellers and buyers. Moreover, the substantially higher frequency of "agree" compared to "deny" and "disagree" might indicate a tendency towards successful negotiations in the dataset.

Domain	# Actions ($\mid \mathcal{A} \mid$)	Cases (Train/Dev/Test)
Movie	8	190/121/161
Music	11	139/109/120
POI	5	96/42/65

Table 9: The detailed statistics regarding different domains in DuRecDial 2.0. The statistics are reported after the data preprocessing step.

A.13 Detailed Prompting Methods for Response Generation

A.13.1 Negotiation Dialogues

For a given action a, we initially map the action a to a textual description, consistent with prior research (Deng et al., 2023b). This mapping is detailed in Table 11. Subsequently, we generate a response using the prompting scheme illustrated in Figure 16. Moreover, existing dialogue policy learning approaches focus solely on strategy prediction, which confounds the evaluation of response generation models. This is due to the proposed

[7] https://github.com/cs-holder/DPDP

DuRecDial 2.0				
Strategy	**Amount**	**Movie**	**Music**	**POI**
Greetings	11,027	✓	✓	✓
Ask about weather	4,393	✗	✓	✓
Play music	10,026	✗	✓	✗
Q/A	6,072	✓	✓	✗
Music on demand	1,692	✗	✓	✗
Movie recommendation	14,807	✓	✓	✗
Chat about stars	16,276	✓	✓	✗
Say goodbye	12,819	✓	✓	✓
Music recommendation	13,170	✓	✓	✗
Ask about date	2,401	✓	✓	✗
Ask questions	2,100	✓	✓	✗
POI recommendation	5,448	✗	✗	✓
Food recommendation	4,465	✗	✗	✓
Craigslist Bargain				
greet	1,727			
inquire	2,102			
inform	416			
propose	1,085			
counter	2,876			
counter-noprice	1,201			
confirm	506			
affirm	770			
deny	320			
agree	843			
disagree	97			

Table 10: The detailed statistics regarding the amounts of dialogue strategies in the original DuRecDial 2.0 and Craigslist datasets.

price being deduced by the response model, rather than a separate planning component. Therefore, in addition to the strategy, we also predict the price within the generated response. However, direct price prediction is challenging due to its continuous nature. To simplify this task, we discretize the price range between the buyer's and seller's prices into B bins and predict the bin containing the target price. Formally, we represent an action a as a tuple (g, b), where g denotes a strategy (as listed in Table 10) and b represents the predicted price bin, where $b \in [0, ..., B-1]$. In the default configuration, we set $B = 5$. Consequently, given the user and the seller's desired prices (denoted as p_{buyer} and p_{seller}, respectively), the estimated price can be computed using the following formulation:

$$\text{price} = p_{\text{buyer}} + b * \frac{p_{\text{seller}} - p_{\text{buyer}}}{B}, \qquad (7)$$

In our experiments, there are 3 strategies combined with predicted prices, namely "propose", "counter", and "agree".

A.13.2 Recommendation Dialogues

Similar to the approach utilized for negotiation, we first convert each action a to a textual description, as detailed in Table 12. Then we prompt LLM to generate a response, using the method illustrated in Figure 17. Moreover, for recommendation-centric actions $\mathcal{A}_{\text{rec}}$, we generate the response, containing the target item v.

A.14 Detailed Prompting for User Simulators

In this section, we provide the detailed prompting methods utilized to instantiate the user simulators for negotiation and recommendation dialogues.

A.14.1 Negotiation Dialogues

In Figure 14, we illustrate the prompting method for the user simulator for the **Craigslist Bargain** dataset. Following previous works (Deng et al., 2023b; He et al., 2024a; Zhang et al., 2024), we prompt LLM to role-play a seller engaged in a bargaining dialogue, with the objective of selling a product at a specified **seller-desired price.**

A.14.2 Recommendation Dialogues

In Figure 15, we illustrate the prompting strategies employed for the **DuRecDial 2.0** dataset. Specifically, we prompt LLM, looking for an item, to simulate a user engaged in a recommendation dialogue. Furthermore, to enhance the realism of simulated user behavior, we incorporate a personal profile into the user simulator prompt. In particular, we retrieve background knowledge regarding users from the background dataset, such as their accepted and rejected items, and prompt an LLM to generate a comprehensive user profile.

A.15 Instructions for Human Evaluation

In this section, we provide instructions for human evaluation. Specifically, we invite two annotators to score conversations to score dialogues across three dimensions: **Deal Achievement** and **Negotiation Equity**, and **Buyer's Benefit**. During the evaluation process, we provide the annotator with the task background, containing the product name as well as the buyer and seller's desired prices. Then we ask the annotators to answer the following questions:

- **Deal Achievement:** *Which conversation ends with a common deal between the buyer and the seller ?*

- **Negotiation Equity:** *Given the seller and the buyer's desired prices, which conversation ends with a fairer deal for both buyer and seller ?*

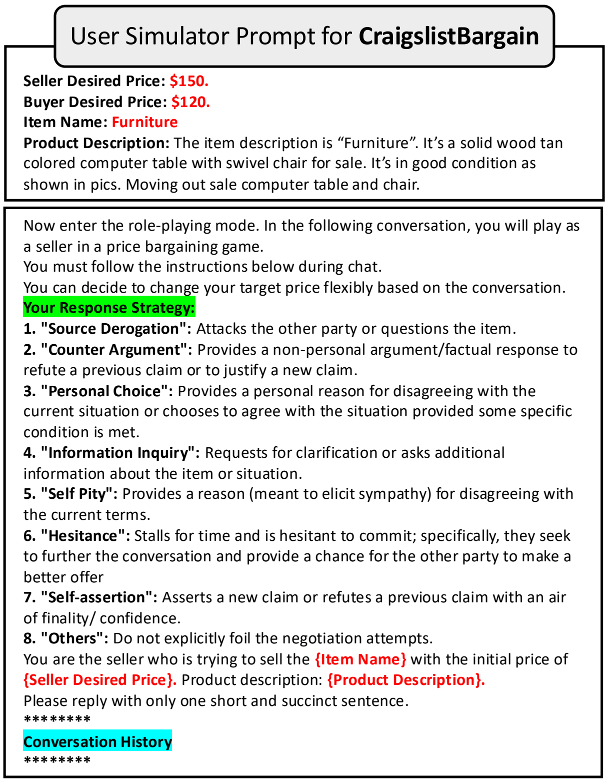
Strategy	Natural Language Form
greet	Please say hello or chat randomly.
inquire	Please ask any question about product, year, price, usage, etc.
inform	Please provide information about the product, year, usage, etc.
propose, {price}	Please propose the price of {price}.
counter, {price}	Please counter the seller with the price of {price}.
counter-noprice	Please propose a vague price by using comparatives with an existing price.
confirm	Please ask a question about the information to be confirmed.
affirm	Please give an affirmative response to a confirm.
deny	Please give a negative response to a confirm.
agree, {price}	Please agree with the price of {price}.
disagree	Please disagree with the proposed price.

Table 11: The strategies and their corresponding textual description utilized for negotiation dialogues in our work.

Strategy	Natural Language Form
Ask about weather	Please provide information about the weather.
Play music	Please select an appropriate song from your given topic set and reply that song is playing.
Music recommendation, {target item}	Please recommend the song {target item} to the user
Q&A	Please answer questions asked by the user
Chat about stars	Please select an appropriate movie star from your given topic set and provide information about the movie star
Music on demand	Please select an appropriate song from your given topic set and reply that song is suitable for the user demand
Movie recommendation, {target item}	Please recommend the movie {target item} to the user.
Say goodbye	Please say goodbye to the user.
Ask about date	Please provide information regarding date.
Ask questions	Please select an appropriate topic from your given topic set and ask questions regarding that topic
Greetings	Please say hello or chat randomly.
POI recommendation, {target item}	Please recommend the restaurant {target item} to the user.
Food recommendation, {target item}	Please recommend the food {target item} to the user

Table 12: The strategies and their corresponding textual description utilized for recommendation dialogues in our work.

- **Buyer's Benefit:** *Given the buyer's desired price, which conversation ends with a deal price which is more beneficial for the buyer?*

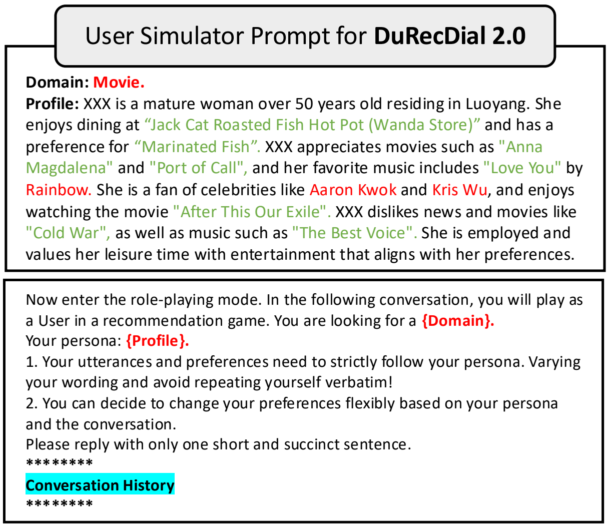


User Simulator Prompt for **CraigslistBargain**

Seller Desired Price: $150.
Buyer Desired Price: $120.
Item Name: Furniture
Product Description: The item description is "Furniture". It's a solid wood tan colored computer table with swivel chair for sale. It's in good condition as shown in pics. Moving out sale computer table and chair.

Now enter the role-playing mode. In the following conversation, you will play as a seller in a price bargaining game.
You must follow the instructions below during chat.
You can decide to change your target price flexibly based on the conversation.
Your Response Strategy:
1. **"Source Derogation":** Attacks the other party or questions the item.
2. **"Counter Argument":** Provides a non-personal argument/factual response to refute a previous claim or to justify a new claim.
3. **"Personal Choice":** Provides a personal reason for disagreeing with the current situation or chooses to agree with the situation provided some specific condition is met.
4. **"Information Inquiry":** Requests for clarification or asks additional information about the item or situation.
5. **"Self Pity":** Provides a reason (meant to elicit sympathy) for disagreeing with the current terms.
6. **"Hesitance":** Stalls for time and is hesitant to commit; specifically, they seek to further the conversation and provide a chance for the other party to make a better offer
7. **"Self-assertion":** Asserts a new claim or refutes a previous claim with an air of finality/ confidence.
8. **"Others":** Do not explicitly foil the negotiation attempts.
You are the seller who is trying to sell the **{Item Name}** with the initial price of **{Seller Desired Price}.** Product description: **{Product Description}.**
Please reply with only one short and succinct sentence.
****
Conversation History
****

Figure 14: The user simulator prompt for **Craigslist Bargain** dataset.

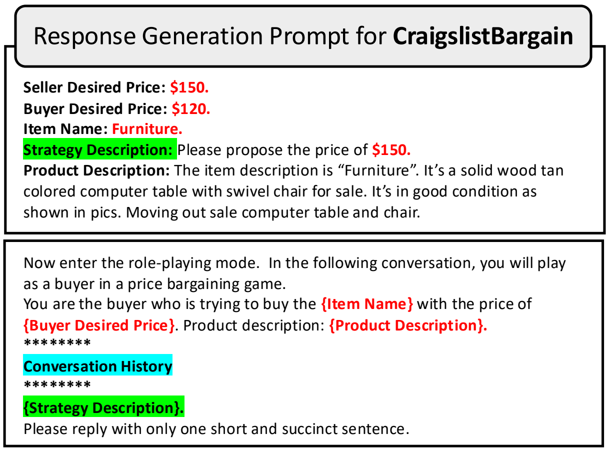


User Simulator Prompt for **DuRecDial 2.0**

Domain: Movie.
Profile: XXX is a mature woman over 50 years old residing in Luoyang. She enjoys dining at "Jack Cat Roasted Fish Hot Pot (Wanda Store)" and has a preference for "Marinated Fish". XXX appreciates movies such as "Anna Magdalena" and "Port of Call", and her favorite music includes "Love You" by Rainbow. She is a fan of celebrities like Aaron Kwok and Kris Wu, and enjoys watching the movie "After This Our Exile". XXX dislikes news and movies like "Cold War", as well as music such as "The Best Voice". She is employed and values her leisure time with entertainment that aligns with her preferences.

Now enter the role-playing mode. In the following conversation, you will play as a User in a recommendation game. You are looking for a **{Domain}.**
Your persona: **{Profile}.**
1. Your utterances and preferences need to strictly follow your persona. Varying your wording and avoid repeating yourself verbatim!
2. You can decide to change your preferences flexibly based on your persona and the conversation.
Please reply with only one short and succinct sentence.
****
Conversation History
****

Figure 15: The user simulator prompt for **DuRecDial 2.0** dataset.

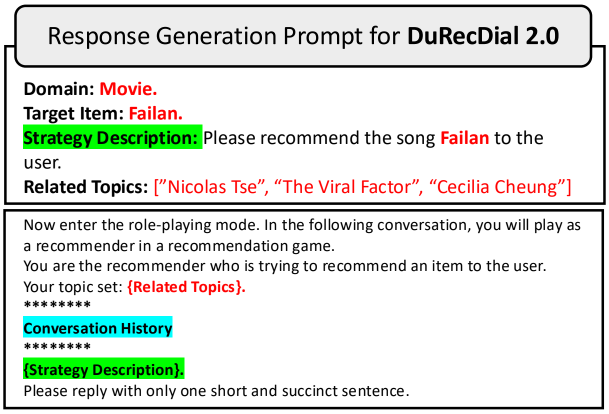


Response Generation Prompt for **CraigslistBargain**

Seller Desired Price: $150.
Buyer Desired Price: $120.
Item Name: Furniture.
Strategy Description: Please propose the price of $150.
Product Description: The item description is "Furniture". It's a solid wood tan colored computer table with swivel chair for sale. It's in good condition as shown in pics. Moving out sale computer table and chair.

Now enter the role-playing mode. In the following conversation, you will play as a buyer in a price bargaining game.
You are the buyer who is trying to buy the **{Item Name}** with the price of **{Buyer Desired Price}**. Product description: **{Product Description}.**
****
Conversation History
****
{Strategy Description}.
Please reply with only one short and succinct sentence.

Figure 16: The response generation prompt for **Craigslist Bargain** dataset.

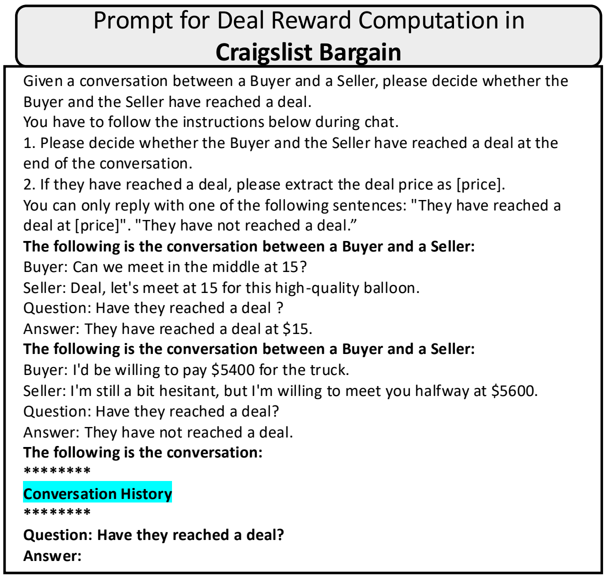


Response Generation Prompt for **DuRecDial 2.0**

Domain: Movie.
Target Item: Failan.
Strategy Description: Please recommend the song **Failan** to the user.
Related Topics: ["Nicolas Tse", "The Viral Factor", "Cecilia Cheung"]

Now enter the role-playing mode. In the following conversation, you will play as a recommender in a recommendation game.
You are the recommender who is trying to recommend an item to the user.
Your topic set: **{Related Topics}.**
****
Conversation History
****
{Strategy Description}.
Please reply with only one short and succinct sentence.

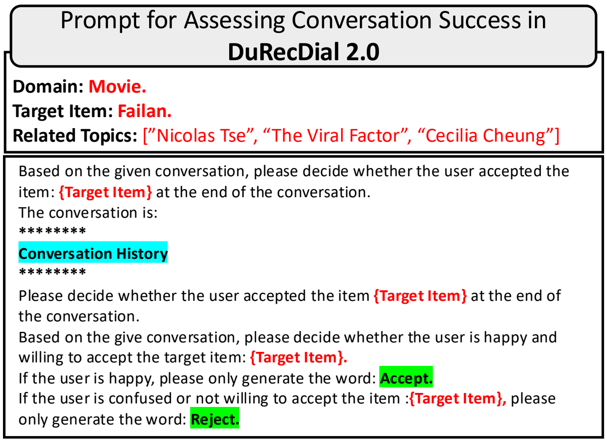
Figure 17: The response generation prompt for **DuRecDial 2.0** dataset.



Prompt for Deal Reward Computation in **Craigslist Bargain**

Given a conversation between a Buyer and a Seller, please decide whether the Buyer and the Seller have reached a deal.
You have to follow the instructions below during chat.
1. Please decide whether the Buyer and the Seller have reached a deal at the end of the conversation.
2. If they have reached a deal, please extract the deal price as [price].
You can only reply with one of the following sentences: "They have reached a deal at [price]". "They have not reached a deal."
The following is the conversation between a Buyer and a Seller:
Buyer: Can we meet in the middle at 15?
Seller: Deal, let's meet at 15 for this high-quality balloon.
Question: Have they reached a deal ?
Answer: They have reached a deal at $15.
The following is the conversation between a Buyer and a Seller:
Buyer: I'd be willing to pay $5400 for the truck.
Seller: I'm still a bit hesitant, but I'm willing to meet you halfway at $5600.
Question: Have they reached a deal?
Answer: They have not reached a deal.
The following is the conversation:
****
Conversation History
****
Question: Have they reached a deal?
Answer:

Figure 18: The prompt for computing the deal reward (r_{deal}) in Craigslist Bargain dataset.



Prompt for Assessing Conversation Success in **DuRecDial 2.0**

Domain: **Movie.**
Target Item: **Failan.**
Related Topics: [”Nicolas Tse”, “The Viral Factor”, “Cecilia Cheung”]

Based on the given conversation, please decide whether the user accepted the item: **{Target Item}** at the end of the conversation.
The conversation is:
****
Conversation History
****
Please decide whether the user accepted the item **{Target Item}** at the end of the conversation.
Based on the give conversation, please decide whether the user is happy and willing to accept the target item: **{Target Item}.**
If the user is happy, please only generate the word: **Accept.**
If the user is confused or not willing to accept the item :**{Target Item},** please only generate the word: **Reject.**

Figure 19: The prompt for assessing the conversation success in the **DuRecDial 2.0** dataset.