

# Experience as Source for Anticipation and Planning: Experiential Policy Learning for Target-driven Recommendation Dialogues

Huy Dao<sup>1</sup>, Yang Deng<sup>1</sup>, Khanh-Huyen Bui<sup>2</sup>, Dung D. Le<sup>3</sup>, Lizi Liao<sup>1</sup>

<sup>1</sup>Singapore Management University

<sup>2</sup>FPT Software AI Center, <sup>3</sup>College of Engineering and Computer Science, VinUniversity  
qh.dao.2023@phdcs.smu.edu.sg, {ydeng, lzliao}@smu.edu.sg,  
huyenbk2@fpt.com, dung.ld@vinuni.edu.vn,

## Abstract

Target-driven recommendation dialogues present unique challenges in dialogue management due to the necessity of anticipating user interactions for successful conversations. Current methods face significant limitations: (I) inadequate capabilities for conversation anticipation, (II) computational inefficiencies due to costly simulations, and (III) neglect of valuable past dialogue experiences. To address these limitations, we propose a new framework, Experiential Policy Learning (EPL), to enhance such dialogues. Specifically, EPL embodies the principle of *Learning From Experience*, facilitating *anticipation* with an experiential scoring function that estimates dialogue state potential using similar past interactions stored in long-term memory. Moreover, to demonstrate its flexibility, we introduce Tree-structured EPL (T-EPL) as one possible training-free realization with Large Language Models (LLMs) and Monte-Carlo Tree Search (MCTS). In particular, T-EPL assesses past dialogue states with LLMs while utilizing MCTS to achieve hierarchical and multi-level reasoning. Extensive experiments on two published datasets demonstrate the superiority and efficacy of T-EPL. Code and data are available here <sup>1</sup>.

## 1 Introduction

Conversational recommender systems (CRSs) (Li et al., 2018; Zhou et al., 2020; Ma et al., 2021; Li et al., 2022; Wu et al., 2022; Liao et al., 2023; Dao et al., 2024) aim to provide suitable recommendations through interactive multi-turn dialogues. Traditional CRSs typically employ a reactive approach, identifying user interests during the conversation and mapping these preferences to a set of candidate items. While effective in certain contexts, these passive systems fall short in their ability to proactively guide users towards specific items,

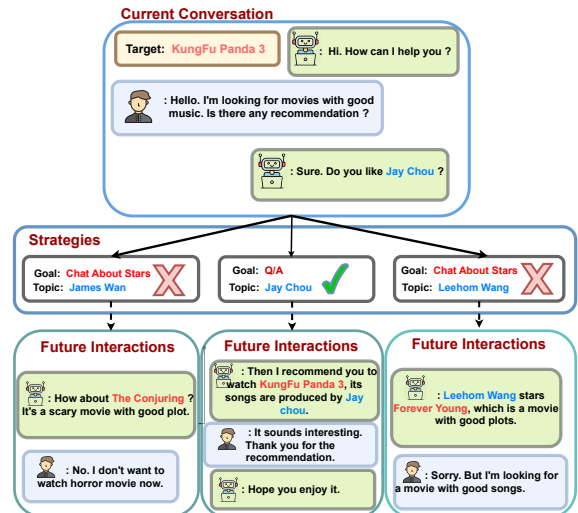


Figure 1: In target-driven recommendation dialogues, foresight interactions are key to guiding conversations toward the desired item.

a capability crucial for promoting new products and increasing sales revenue (Dao et al., 2023). Recently, target-driven CRS models (Wang et al., 2022, 2023b,a) have emerged as a more dynamic alternative. These models leverage various dialogue strategies (e.g., chit-chat, question-answering, etc.) to engage users and foster their interest in target items, aiming to recommend them when the timing and context are most appropriate.

Despite their effectiveness, existing target-driven CRS models exhibit a notable shortcoming: the inability to anticipate future user-system interactions, as they focus solely on individual next-turn evaluations. In target-driven recommendation dialogues, anticipating conversational trajectories toward specific target items is essential for successful interactions. Hence, the integration of foresight user-system interactions has the potential to enhance dialogue planning capabilities (He et al., 2024). For instance, as illustrated in Figure 1, by considering potential future interactions, the dialogue strategy (Q/A, Jaychou) is apparently the optimal choice since it effectively steers the conversation toward

<sup>1</sup><https://github.com/huyquangdao/EPL>

the desired target *KungFu Panda 3*. Previous works attempt to leverage simulated user interactions generated by LLMs and employ either Reinforcement Learning (RL) (Deng et al., 2023a) or open-loop Monte-Carlo Tree Search (MCTS) (Yu et al., 2023) to enhance their dialogue policies. However, these approaches encounter two main challenges. First, fixed pre-trained policies generated during training may not adapt well to new user interactions during deployment. Second, constant access to LLMs for state evaluations can lead to computational inefficiencies. Notably, almost all prior work fails to leverage newly obtained interactions during inference to further enhance their performance.

Recent advancements have demonstrated the benefits of experience reflection in various domains, including multimodal response generation (Ye et al., 2022), decision-making (Shinn et al., 2023), and recommendation (Lin et al., 2023). These approaches often utilize analogous examples or historical trials to deepen understanding of current inputs. For example, in recommender systems, similar users tend to exhibit similar preferences and behaviors (Lin et al., 2023), suggesting that past interactions with such users could provide valuable insights for ongoing conversations. Furthermore, these similar interactions can reduce the need for expensive simulations to estimate future interactions. Inspired by these observations, we leverage similar past dialogue states from previous sessions to enhance target-driven recommendations, addressing key research questions: (1) How can we integrate experienced interactions into **future anticipation**? (2) How can we establish adaptable policies for **new interactions on the fly**?

To address these challenges, we introduce Experiential Policy Learning (EPL), a novel dialogue policy framework for target-driven CRSs. Our proposed EPL approximates a target-driven scoring function for future anticipation by utilizing similar past interactions and their associated values rather than relying on expensive online trials, such as rollout simulations. Specifically, given a dialogue state, EPL retrieves the  $k$  most similar past interactions and their corresponding potential values from a dedicated memory structure. These values are then aggregated to approximate the potential value for the given state. To demonstrate the flexibility of the EPL framework, we introduce Tree-structured EPL (T-EPL) as one possible realization. T-EPL is a training-free implementation that leverages LLMs to assess dialogue states’ potentials

while integrating our target-driven scoring function into an MCTS algorithm for state value estimation, thus reducing the need for costly LLM-based evaluations. Extensive evaluations demonstrate the superiority of our method over existing approaches. To sum up, our contributions are threefold:

- We propose a novel dialogue policy learning framework, named EPL, which incorporates past interactions for future anticipation in planning.
- We introduce T-EPL, a training-free realization of EPL, utilizing LLMs and MCTS, which can quickly adapt to newly encountered interactions.
- We conduct interactive evaluations on two published datasets. Extensive experiments demonstrate the superiority of our T-EPL against SOTA approaches in both performance and efficiency.

## 2 Related Work

**Target-driven Proactive Dialogue Systems.** Recent efforts (Liu et al., 2020; Zhang et al., 2021; Liu et al., 2021; Wang et al., 2022; Deng et al., 2023b; Wang et al., 2023a; Dao et al., 2023) have increasingly focused on developing proactive dialogue systems designed to guide conversations toward predefined targets, such as negotiation (Deng et al., 2023a) and recommendation (Dao et al., 2023). Within this target-driven paradigm, establishing effective dialogue policies to achieve specific outcomes has become a crucial area of research. Several approaches have been proposed, including Brownian-motion dialogue planning (Wang et al., 2023b) and long-short term strategic balancing (Dao et al., 2023). Despite their contributions, these methods lack the ability to incorporate foresight user-system interactions, which are essential for conversation anticipation.

To address this limitation, Deng et al. (2023a) utilized LLM-generated interactions to fine-tune a pre-trained policy with Reinforcement Learning (RL). However, the static nature of their pre-trained policy limits its adaptability to newly encountered situations. Alternatively, Yu et al. (2023) proposed using open-loop Monte-Carlo Tree Search (MCTS) for future interaction estimation, but this approach suffers from computational inefficiencies due to costly LLM-based evaluations. Furthermore, the aforementioned methods do not leverage valuable interactions during inference, which could significantly enhance their performance. In contrast to existing methods, we introduce Experiential Policy

Learning (EPL), a novel dialogue policy method that integrates past interactions into the planning process. By utilizing past user-system interactions, EPL anticipates conversational trajectories and plans more effectively.

**Reflecting on Experience.** Recent research has demonstrated the application of *Reflecting on Experience* in various tasks. For example, Ye et al. (2022) proposed a method to retrieve similar dialogues from training data to enhance the quality of generated responses in multimodal dialogue systems (Liao et al., 2018). Similarly, Lin et al. (2023) focused on extracting information from dialogues with similar users to gain a better understanding of the current user’s preferences. From a different perspective, Shinn et al. (2023) introduced a reflection framework that improved performance in various tasks, such as decision-making, programming, and reasoning. Such a framework utilizes results from past trials to enhance predictions in current ones. Inspired by these, we propose a novel dialogue policy method that utilizes similar past interactions to improve future anticipation, hence boosting performance in target-driven recommendation dialogues.

### 3 Preliminaries

#### 3.1 Markov Decision Process Formalization

In this work, we utilize Markov Decision Process (MDP) defined by a tuple of 5 elements, namely  $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, T, \gamma)$ , including a state space  $\mathcal{S}$ , an action space  $\mathcal{A}$ , and a reward function  $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ , a transition function  $T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$  and a discounted factor  $\gamma$ . Additionally, we denote by  $a \in \mathcal{A}, s \in \mathcal{S}$  an action and a state, respectively. We instance the state at the  $t$ -th conversation turn as  $s_t = (h_{\leq t}, a_{\leq t})$  where  $h_{\leq t}, a_{\leq t}$  are the dialogue context, the sequence of previous actions, respectively.

#### 3.2 Target-driven Recommendation Dialogues

In target-driven recommendation dialogues (Dao et al., 2023), we aim to recommend a pre-specified item  $v \in \mathcal{V}$  ( $\mathcal{V}$  is the set of all targeted items) to the user via multi-turn conversations. Following existing works, we decompose this task into two sub-tasks, described as follows:

**Dialogue Strategy Planning:** Given the current state  $s_t$ , we aim to produce the next action  $a_{t+1}$  so that we can engage the user interest while steering the conversation towards the targeted item  $v$ .

**Response Generation:** Given the state  $s_t$ , the target item  $v$  and the predicted action  $a_{t+1}$ , the system generate response  $y_{t+1}$  to the user.

### 3.3 Interactive Conversations with LLMs.

Following (Deng et al., 2023a), we conduct an interactive evaluation with a user simulator based on LLMs. Specifically, we prompt an LLM to adopt a user’s role seeking a recommendation. Given a target item denoted by  $v$ , the system interacts with the simulator for a predefined number of turns. Upon conversation termination, a separate LLM is employed to assess whether the generated dialogue resulted in a successful recommendation. The prompting schemes used for the user simulator and LLM-based assessment are detailed in Appendix Sections A.12 and 10, respectively.

## 4 Methodology

First, we describe our proposed Experiential Policy Learning (EPL) in sections 4.1. Subsequently, we introduce Tree-structured EPL (T-EPL), a training-free realization of the EPL framework in Section 4.2. Figure 2 illustrates our proposed T-EPL.

### 4.1 Experiential Policy Learning

**Target-driven Scoring Function.** Since the outcome of a conversation can be either *success* or *fail*. Hence, we model the outcome  $r$  of a conversation using a binary random variable, accepting two possible values  $\mathcal{L} = \{success, fail\}$ . Given a state  $s$  and a target item  $v$ , one could estimate  $F(s, v)$ , which is the potential value of  $s$  and  $v$ , as follows:

$$\begin{aligned} F(s, v) &= \mathbb{E}_{r \sim \mathbf{P}(r|s, v)} [f(r)], \\ &= \sum_{r \in \mathcal{L}} \mathbf{P}(r|s, v) f(r), \end{aligned}$$

where  $f(r) : \mathcal{L} \rightarrow \mathbb{R}$  is a scalar-valued function that maps an outcome to its corresponding scalar value and  $\mathbf{P}(r|s, v)$  is the probability of the outcome  $r$  given the state  $s$  and the target item  $v$ . Estimating  $\mathbf{P}(r|s, v)$  can be challenging since the state  $s$  might capture limited information to predict conversation outcomes. A potential solution is to re-formalize the function  $F(s, v)$  as follows:

$$F(s, v) = \sum_{r \in \mathcal{L}} f(r) \sum_{c \sim \mathbf{P}(c|s, v)} \mathbf{P}(r|c, s, v) \mathbf{P}(c|s, v),$$

where  $c$  can be interpreted as a dialogue continuation, encompassing subsequent user-system interactions. Given a continuation  $c$ , the outcome might

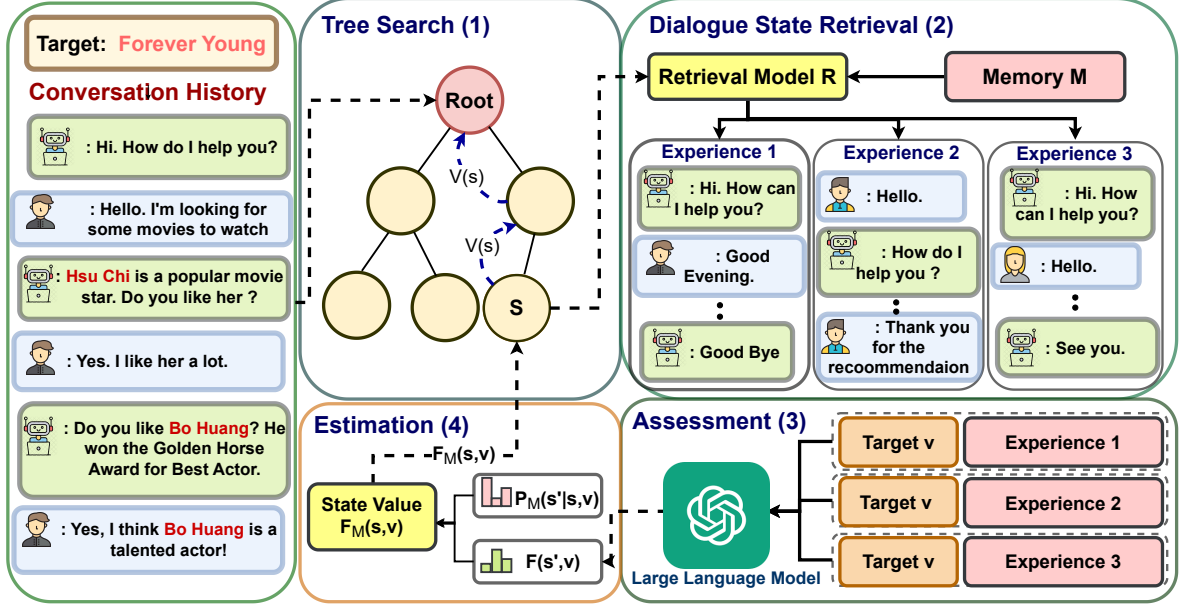


Figure 2: The illustration of our proposed T-EPL algorithm for target-driven recommendation dialogues. Specifically, T-EPL employs MCTS to construct a **Search Tree** and evaluate node’s potential value using an experiential target-driven scoring function (Section 4.1). In particular, the **Retrieval** phase refers to E.q. 3. The **Assessment** and **Estimation** phases are mathematically described by E.q. 2 and E.q. 1, respectively.

be directly determined (e.g.,  $\mathbf{P}(r = 1|c, s, v) = 1 \Leftrightarrow v \in c$ ). However, this reformulation also introduces a new challenge, which is probability  $\mathbf{P}(c|s, v)$ . While explicitly computing or modeling this probability is computationally intractable, one can approximately draw samples from this distribution by using online rollout simulations (Zhang et al., 2023; Zhou et al., 2023). Nonetheless, this approach still suffers from computational inefficiency due to simulating numerous completed dialogues at inference time.

**An Experiential Approximation.** Since similar users tend to exhibit similar preferences and behaviors (Lin et al., 2023), leveraging analogous experienced interactions can provide valuable guidance for the current conversation. Hence, we propose an approximated scoring function  $F_{\mathcal{M}}(s, v)$ , using similar past interactions, as follows:

$$F_{\mathcal{M}}(s, v) = \sum_{r \in \mathcal{L}} f(r) \cdot \sum_{c' \sim \mathbf{P}_{\mathcal{M}}(c'|s, v)} \mathbf{P}(r|c', s, v) \mathbf{P}(c'|s, v),$$

where  $\mathcal{M} = \{(s', c', v', F(s', v'))\}$  is a memory, storing tuples of experienced state  $s'$ , corresponding continuation  $c'$ , target item  $v'$  and a target-driven assessment score  $F(s', v')$ . Since  $(s', c')$  forms a completed conversation, we could regard  $s'$  as a proxy for  $c'$ . Re-arranging the order of the

summation, we obtain the following formulation:

$$F_{\mathcal{M}}(s, v) = \sum_{(s', c') \sim \mathbf{P}_{\mathcal{M}}(s'|s, v)} \mathbf{P}(s'|s, v) \cdot \sum_{r \in \mathcal{L}} f(r) \mathbf{P}(r|c', s, v),$$

By only considering  $k$  most similar states  $s' \in \mathcal{M}_k$  for a given state  $s$ , we assume that  $\mathbf{P}(r|c', s, v) \approx \mathbf{P}(r|c', s', v)$ . This avoids computing multiples  $\mathbf{P}(r|c', s, v)$ , which might be costly during inference. Hence, we obtain the following formulation:

$$F_{\mathcal{M}}(s, v) \approx \sum_{s' \in \mathcal{M}_k} \mathbf{P}(s'|s, v) F(s', v), \quad (1)$$

$$= \mathbb{E}_{s' \sim \mathbf{P}_{\mathcal{M}_k}^k(s'|s, v)} [F(s', v)],$$

where we assume that transitions are deterministic under a policy  $\pi(a|s)$ . That means  $\mathbf{P}(r|c', s', v') = \mathbf{P}(r|s', v')$  for a completed conversation  $(s', c')$  and  $F(s', v) = 0, \forall (s', c', v) \notin \mathcal{M}$ . This formulation bypasses the need for rollout simulations while facilitating a fast approximation of the target-driven function using similar past interactions.

## 4.2 Tree-structured EPL (T-EPL)

To adapt EPL with newly encountered interactions on the fly, it necessitates a training-free implementation of E.q. 1. Moreover, planning is often formalized as a hierarchical, multi-level process. Hence, in this section, we introduce a training-free realization of EPL using LLMs, a dense document

---

**Algorithm 1** Tree-structured EPL (T-EPL)

---

**Input:** The current state  $s_t$ , UCT parameter  $c$ , Number of simulations  $n$ , Prior policy  $\pi_{prior}(\cdot|\cdot; \theta)$ , Memory  $\mathcal{M}$ , Number of retrieved experiences  $k$ , Target  $v$

**Output:** Next action  $a_{t+1}$

```
1: root  $\leftarrow s_t$ ;  
2: for  $i \leftarrow 1$  to  $n$  do  
3:    $s \leftarrow \text{root}$ ;  
4:   while  $s$  is not a leaf node do ▷ Selection;  
5:      $s \leftarrow \text{UCT}(\text{children}(s), \pi_{prior}, c)$ ;  
6:   end while  
7:    $\hat{\mathcal{A}}_{t+1} \sim \pi_{prior}(a|s; \theta)$ ;  
8:   for  $\hat{a}_{t+1} \in \hat{\mathcal{A}}_{t+1}$  do ▷ Expansion  
9:      $\hat{s}_{t+1} \leftarrow T(s, \hat{a}_{t+1})$ ;  
10:    if  $\hat{s}_{t+1}$  not in tree then  
11:       $\text{children}(s) \leftarrow \text{children}(s) \cup \hat{s}_{t+1}$ ;  
12:       $s \leftarrow \hat{s}_{t+1}$ ;  
13:      Break;  
14:    end if  
15:  end for  
16:   $\mathcal{M}_k \leftarrow \text{Retrieval}(\mathcal{M}, s, k)$ ; ▷ Estimation  
17:   $F_{\mathcal{M}}(s, v) \leftarrow f_{score}(s, \mathcal{M}_k, v)$ ;  
18:   $V_s \leftarrow V_s^{\mathcal{M}_k}$ ; ▷ BackPropagation;  
19:  while  $s$  is not the root do  
20:     $p \leftarrow \text{parent}(s)$ ;  
21:     $V_p \leftarrow \max(V_p, V_s)$ ;  
22:     $N(s) \leftarrow N(s) + 1$ ;  
23:     $s \leftarrow \text{Parent}(s)$ ;  
24:  end while  
25: end for  
26: Return: The action  $\hat{a}_{t+1}$  of the root state  $s_t$  that has the highest value  $R(s_t, \hat{a}_{t+1}, \hat{s}_{t+1}) + V_{\hat{s}_{t+1}}$ ;
```

---

retrieval model, and the MCTS algorithm, as described as follows:

**MCTS-guided Tree Search:** We leverage MCTS’s procedure to establish a tree search for facilitating hierarchical and multi-step reasoning. Existing MCTS-based algorithms rely on either expensive rollout simulations (Zhang et al., 2023) or seamless LLM evaluations (Yu et al., 2023). In this work, we propose to equip  $F_{\mathcal{M}}(s, v)$ , estimated from similar past interactions, as the state value function for MCTS to enhance both its planning capability and efficiency. The detailed procedure is described in Algorithm 1 and Appendix A.1.

**LLM-based Target-driven Assessment:** To establish a training-free realization of EPL, we leverage Llama 2 to assess how successful a conversation  $(s', c')$  regarding a target item  $v'$ . Moreover, we also add a small term to penalize lengthy trajectories. Specifically, the assessment score  $F(s', v')$  is computed as:

$$\frac{1}{T} \sum_{i=1}^T \mathcal{V}_r(\text{LLM}(\mathcal{P}(s', e', v'))) + \lambda \cdot e^{-\frac{l_{c'}}{\alpha}}, \quad (2)$$

where  $\lambda, \alpha$  are hyper parameters and  $l_{c'}$  is the length of the continuation  $c'$ , measured in the num-

ber of turns.  $\mathcal{V}_r(\cdot)$  maps a textual output to a scalar value  $r \in \mathcal{R}$  while  $\mathcal{P}(\cdot)$  maps the tuple of  $(s', c', v')$  into the input prompt for the LLM. We prompt the LLM for  $T$  times with the temperature  $\rho = 1.1$ . Furthermore, we leverage *GPT-3.5-Turbo* to instance the LLM component.

**Dense Retrieval of Dialogue States:** In this work, we model the probability distribution  $\mathbf{P}_{\mathcal{M}}(s'|s, v)$  with a retrieval model  $f_{rev}(s'|s; \theta_{rev})$ . Specifically, we compute the similarity score between the current state  $s$  and an experienced state  $s'$  in the memory  $\mathcal{M}$  by using the following formulation:

$$\mathbf{P}_{\mathcal{M}}(s'|s) = \frac{\exp(\phi(s')^T \phi(s, v))}{\sum_{s'' \in \mathcal{M}} \exp(\phi(s'')^T \phi(s, v))}, \quad (3)$$

where  $\phi$  is an encoder function that maps dialogue states to their corresponding high-dimensional representations. In this work, we establish our retrieval model with the pre-trained encoder *all-MiniLM-L6-v2* from the Sentence Transformers package<sup>2</sup>.

**Memory Construction.** In real-world applications, a recommender system might need to interact with users seamlessly and dynamically update its database with new interactions for future uses. Hence, we adopt a similar scenario where a prior policy  $\pi_{prior}(a|s; \theta)$  interacts with the simulator multiple times and updates the memory with new conversations. For each generated dialogue, we utilize Eq 2 to compute its assessment score. Later, we break it down into dialogue states and continuations and store them in the memory. In this work, we leverage Faiss<sup>3</sup> to instance the memory  $\mathcal{M}$ .

## 5 Experiments

### 5.1 Experimental Setup

**Datasets** For experiments, we leverage two published datasets. The first dataset, **DuRecDial 2.0** (Liu et al., 2021), encompasses conversations across multiple domains. The second dataset, **INSPIRED** (Hayati et al., 2020), focuses on movie recommendation scenarios. We provide detailed statistics on these datasets in Table 1 and Appendix A.3. Following previous approaches (Dao et al., 2023), we regard the targeted item as the one the user accepts at the end of each conversation.

**Baselines** In this work, we compare our proposed **T-EPL** with various dialogue policy methods, including **BERT** (Devlin et al., 2019), **TCP** (Wang

<sup>2</sup><https://www.sbert.net/>

<sup>3</sup><https://github.com/facebookresearch/faiss>

	DuRecDial 2.0	INSPIRED
# convs	16.5K	1001
# utterances	255K	35,811
# goals	13	14
# topics	646	1169
# target items	471/285/376	368/42/55
domains	Movie/Music/Food/POI	Movie

Table 1: The detailed statistics of datasets.

et al., 2022), **UNIMIND** (Deng et al., 2023b)), **COLOR** (Wang et al., 2023b), **RTCP** (Dao et al., 2023), **GDP-Zero** (Yu et al., 2023) and **PPDPP** (Deng et al., 2023a). Moreover, we also report the performance of vanilla Monte-Carlo Tree Search with rollouts (i.e., **MCTS**). For MCTS-based approaches such as MCTS, GDP-Zero, and T-EPL, we utilize RTCP as their prior policies for a fair comparison. The implementation details of T-EPL and additional information about baselines can be found in Appendix A.9 and A.10, respectively.

**Evaluation Metrics** We employ both automatic and human evaluations to assess model performance. For automatic evaluation, we utilize two primary evaluation metrics: objective success rate  $\mathbf{Obj}_{SR}$  and subjective success rate  $\mathbf{Subj}_{SR}$ . While  $\mathbf{Obj}_{SR}$  measures whether the generated response contains the target item,  $\mathbf{Subj}_{SR}$  determines if the LLM-based assessment score of a generated conversation surpasses a pre-defined threshold  $\epsilon$ . Moreover, we also report the average number of turns (**Avg. T**) required for recommending the target item. For efficiency comparison, we report the approximated number of API calls (**#APIC**) (in terms of Big-O notation) consumed by each model at inference time. For human evaluation, we randomly sample 20 dialogues generated by each model and present them to two annotators. Following (Dao et al., 2023), we leverage **Satisfaction** and **Coherency** as evaluation metrics. We report T-EPL’s win and loss rates against baselines for these metrics. Instructions provided to human annotators for evaluation are included in the Appendix (Section A.15). We also report the inter-annotator agreement using Fleiss’ Kappa (McHugh, 2012).

## 5.2 Main Findings

We present the main empirical results of the automatic evaluation in Table 2.

**Objective versus Subjective Metrics.** First, the results reveal a consistent and substantial difference between the objective and subjective success rates (i.e.  $\mathbf{Obj}_{sr}$  and  $\mathbf{Subj}_{sr}$ ) across all methods.

This observation is reasonable as the user might reject the recommended target item. Consequently, relying solely on  $\mathbf{Obj}_{SR}$ , as in prior studies (Wang et al., 2023b; Dao et al., 2023), may lead to an over-estimation of the effectiveness of target-driven recommendation systems in real-world applications.

**Generative versus Predictive Policies.** Secondly, we experience that predictive policies (e.g., BERT, RTCP) achieve superior performance on the DuRecDial 2.0, whereas generative counterparts (e.g., TCP and COLOR) are more effective on the INSPIRED. This is expected since INSPIRED has a significantly larger action space compared to DuRecDial 2.0. Predictive policies often struggle to handle large action spaces, whereas generative policies are less susceptible to this limitation. Further analyses regarding these two categories of dialogue policies are presented in Appendix A.4.

**Performance Comparison against Baseline Methods.** Our proposed T-EPL algorithm consistently outperforms all existing target-driven dialogue policies (denoted by  $*$ ) across various datasets and evaluation metrics. This superiority can be attributed to T-EPL’s capability to effectively utilize similar past interactions stored within the memory  $\mathcal{M}$ , leading to enhanced dialogue planning abilities. Interestingly, PPDPP exhibits relatively lower performance compared to other baseline methods. We posit that this might be due to PPDPP’s limitation in fine-tuning its pre-trained policy on a limited number of interactions, hindering its generalizability to those encountered during inference. Additionally, computational constraints restrict MCTS to employing a limited number of rollouts, hindering its ability to learn effective policies. Notably, T-EPL demonstrates superior performance to the strong GDP-Zero baseline across several metrics. This suggests that our experiential target-driven scoring function provides accurate assessments of dialogue states compared to the LLM-based estimation employed by GDP-Zero. We further report the results of T-EPL and baseline methods across various domains in Appendix A.7.

**Efficiency Comparison.** Offline policies (denoted by  $*$ ) necessitate minimal API calls to function the user simulator. In contrast, the MCTS baseline incurs a quadratic computational cost w.r.t conversation length due to the costly rollouts. Similarly, GDP-Zero also experiences high API call demands for estimating state values using LLMs.

Model	#APIC	DuRecDial 2.0			INSPIRED		
		Obj <sub>sr</sub>	Subj <sub>sr</sub>	Avg. T(↓)	Obj <sub>sr</sub>	Subj <sub>sr</sub>	Avg. T(↓)
BERT* (Devlin et al., 2019)	$\mathcal{O}(TH)$	0.851	0.773	6.045	0.090	0.062	13.647
UNIMIND* (Deng et al., 2023b)	$\mathcal{O}(TH)$	0.720	0.668	6.873	0.163	0.109	13.321
TCP* (Wang et al., 2022)	$\mathcal{O}(TH)$	0.742	0.679	7.091	0.201	0.168	13.178
COLOR* (Wang et al., 2023b)	$\mathcal{O}(TH)$	0.805	0.749	7.067	0.206	<b>0.172</b>	13.283
RTCP* (Dao et al., 2023)	$\mathcal{O}(TH)$	<u>0.877</u>	<u>0.786</u>	<u>5.993</u>	0.136	0.099	13.479
<b>T-EPL</b> * (ours)	$\mathcal{O}(TKH)$	<b>0.904</b>	<b>0.813</b>	<b>5.255</b>	<b>0.218</b>	<b>0.172</b>	<b>13.158</b>
MCTS <sup>‡</sup>	$\mathcal{O}(TKH^2)$	<u>0.867</u>	0.784	5.550	0.137	0.112	13.537
GDP-Zero <sup>‡</sup> (Yu et al., 2023)	$\mathcal{O}(TKH^2)$	0.850	<u>0.825</u>	<b>4.475</b>	<b>0.250</b>	<u>0.217</u>	<u>13.216</u>
PPDPP <sup>‡</sup> (Deng et al., 2023a)	$\mathcal{O}(TH)$	0.667	0.650	7.283	0.150	0.100	13.416
<b>T-EPL</b> <sup>‡</sup> (ours)	$\mathcal{O}(TKH)$	<b>0.900</b>	<b>0.833</b>	<u>5.034</u>	<u>0.225</u>	<b>0.225</b>	<b>13.000</b>

Table 2: Empirical results on the target achievement where T, H, and K are the number of target items, the maximal number of conversation turns, and simulation steps, respectively. For \*, we report the performance on the whole test set. For ‡, due to their high computational cost in API calls, we follow instructions from (Yu et al., 2023) to sub-sample the datasets and evaluate the models accordingly (t-test,  $p < 0.05$ ).

Model	DuRecDial 2.0		INSPIRED	
	Subj <sub>sr</sub>	Avg. T(↓)	Subj <sub>sr</sub>	Avg. T(↓)
T-EPL	0.813	<b>5.255</b>	<b>0.172</b>	<b>13.158</b>
- w/o Len	<b>0.837</b>	5.312	0.145	13.161
- w/o Exp	0.801	5.435	0.136	13.372

Table 3: Ablation study on the target achievement of our T-EPL algorithm (t-test,  $p < 0.05$ ). Descriptions of these variants can be found in Appendix A.10.

T-EPL vs	Stat.		Coh.	
	Win.(%)	Lose.(%)	Win.(%)	Lose.(%)
RTCP	<b>38 %</b>	32 %	<b>27 %</b>	24 %
COLOR	<b>45 %</b>	34 %	<b>21 %</b>	18 %
PPDPP	<b>27 %</b>	19 %	<b>34 %</b>	23 %
GDP-Zero	<b>32 %</b>	29 %	<b>26 %</b>	22 %

Table 4: Human evaluation on DuRecDial 2.0 dataset. The inter-annotator agreement score is 0.69.

While our proposed T-EPL algorithm introduces additional API calls to simulate past interactions, this step can be pre-computed efficiently. Notably, during inference, T-EPL exhibits a linear scaling of API calls with conversation length, comparable to other offline policy models. We report the computational time of each model in Appendix A.8.

### 5.3 Ablation Study

The results of the ablation study are presented in Table 3. We systematically remove each component of the T-EPL algorithm, including the length-penalized term (denoted as *w/o Len*) and the experiential target-driven scoring function (*w/o Exp*). Overall, removing these components leads to decreases in the performance of our algorithm, demonstrating the effectiveness of our contributions. Interestingly, we observe a trade-off between two metrics, namely **Subj<sub>sr</sub>** and **Avg. T**, on DuRecDial 2.0. In particular, **Subj<sub>sr</sub>** is improved, while

**Avg. T** degrades as we remove the length-penalized term. In contrast, on the INSPIRED dataset, removing this component negatively impacts both **Subj<sub>sr</sub>** and **Avg. T**. The reason might be the INSPIRED dataset contains long-horizon conversations. Hence, estimating accurate state value becomes more challenging, and the accumulated error during planning becomes larger as the conversation unfolds. Hence, by penalizing long trajectories, T-EPL will prioritize shorter experienced interactions despite their lower potential values. Furthermore, removing the experiential target-driven function results in significant performance drops on both datasets, emphasizing the importance of experience in improving target-driven planning capabilities.

### 5.4 Human Evaluation

In Table 4, we report the results of the human evaluation. In particular, we report the win and lose rates of T-EPL against RTCP, COLOR, PPDPP, and GDP-Zero. Overall, our algorithm achieves better performance across different metrics compared to baseline methods. In comparison to its backbone policy RTCP, T-EPL further improves the performance on both **Stat.** and **Coh.** aspects. The reason might be that RTCP suffers from the inability to leverage foresight interaction, which limits its planning ability. In contrast, T-EPL leverages experienced interactions stored in the memory, enhancing the predictions from the backbone policy RTCP.

### 5.5 In-depth Analyses

**Frequency of Target Items and Performance Comparison w.r.t Conversation Turns.** In Figure 3, we present a two-part analysis. The left panel depicts the frequency of target item recommenda-

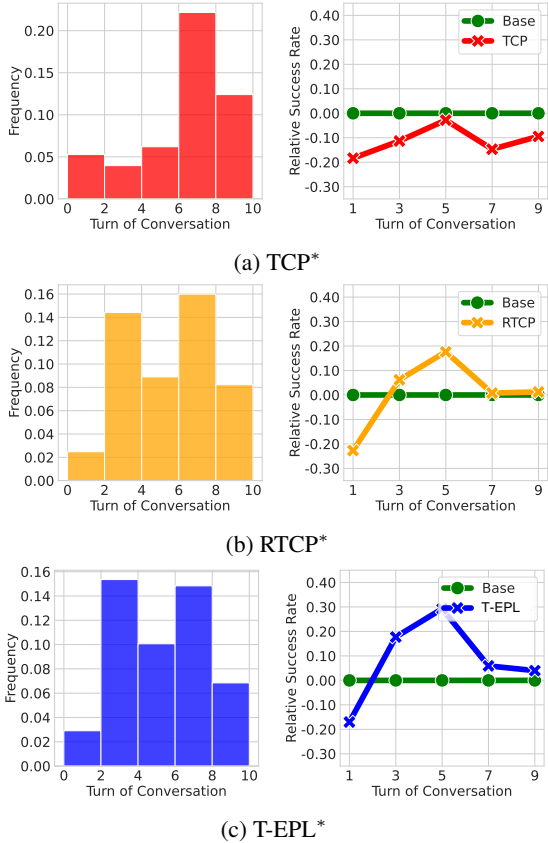


Figure 3: Frequency of target items (left) and relative success rate (right) w.r.t conversation turns of different models. Green lines show the baseline BERT\*. More results can be found in Appendix A.4.

tions across three dialogue policies: TCP, RTCP, and T-EPL. The right panel compares the relative  $\text{Sub}_{sr}$  of these models to the baseline BERT. Our analysis reveals that both RTCP and T-EPL tend to introduce recommendations earlier than TCP. We posit two common dialogue strategies for target-item recommendation. The first strategy prioritizes early recommendations followed by justifications in subsequent turns. Conversely, the second strategy focuses on introducing relevant topics related to the target item in the initial stages, with the actual recommendation occurring later in the dialogue. Qualitative examples provided in Appendix A.14 support this hypothesis. Interestingly, despite employing similar dialogue strategies, T-EPL still achieves better performance gains than its backbone policy RTCP.

### Impact of the Number of Simulation Steps ( $n$ ).

As we integrated our proposed experiential target-driven function into the MCTS, it is desired to examine how other parameters (such as the number of simulations) would affect T-EPL’s performance and efficiency. Figure 4.a shows the results of vary-

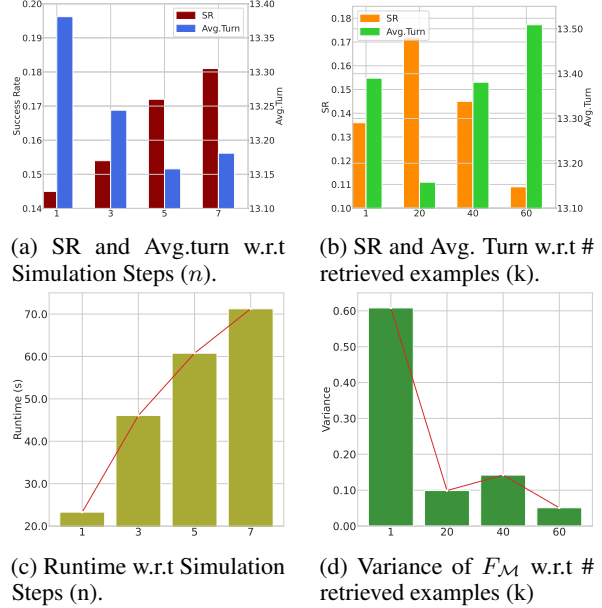


Figure 4: Performance of T-EPL with different values of simulation steps ( $n$ ) and # retrieved examples ( $k$ ).

ing the number of simulation steps ( $n$ ). In this experiment, the number of retrieved interactions  $k$  was fixed at 20, and the model was evaluated with  $n$  values of 1, 3, 5, and 7, respectively. Overall, the performance of T-EPL generally improves with an increasing number of simulation steps. This is expected since a larger value of  $n$  allows the algorithm to construct a more comprehensive search tree. However, it is crucial to acknowledge the inherent trade-off between performance and computational cost. A greater number of simulation steps inevitably incurs a higher computational burden, as evidenced by the increased number of API calls and overall runtime (Figure 4.c). Therefore, selecting an appropriate value for  $n$  necessitates careful consideration to achieve a balance between performance and computational resources.

### Analyses on the Number of Retrieved Interactions ( $k$ ).

Figure 4.b shows the impacts of the number of retrieved interactions  $k$  on T-EPL’s performance. In this experiment, we fix the number of simulation steps ( $n$ ) to 5, while  $k$  is varied across four settings: 1, 20, 40, and 60. The results demonstrate a trend of increasing performance followed by a decrease as the number of retrieved interactions grows. This is expected since a larger value of  $k$  allows the model to consider more interactions, refining policy decisions better. However, a large value of  $k$  also introduces noisy interactions and leads to value estimation saturation (as shown in Figure 4.d). This saturation occurs since each



state is evaluated using increasingly similar sets of interactions, hence hindering model performance.

## 6 Conclusion

In this work, we propose Experiential Policy Learning (EPL), a novel dialogue policy for target-driven recommendation dialogues. Our EPL framework leverages similar past interactions to enhance its dialogue planning capability. Moreover, we introduce Tree-structured EPL, a training-free realization of EPL, utilizing LLMs and MCTS. Through extensive experiments and analyses on two published datasets, we demonstrate the effectiveness of our proposed method in terms of both performance and efficiency against state-of-the-art approaches.

## Limitations

This section explores several potential limitations associated with the proposed T-EPL algorithm: **(1) Memory Availability:** The T-EPL algorithm relies on a memory component. In real-world applications, this memory might not be readily available, potentially requiring additional effort to construct it before deploying the algorithm. **(2) Computational Cost of Interactive LLM Evaluation:** A potential drawback of this work lies in its use of Large Language Models (LLMs) for interactive evaluation. While this approach better reflects practical scenarios, conducting online evaluations using LLMs can incur significant computational costs. **(3) Dependence on Retrieval Models:** T-EPL is also subject to limitations arising from its reliance on pre-trained retrieval models. The performance of T-EPL might be heavily influenced by the quality of retrieved interactions and their corresponding state values.

## Acknowledgements

This research was funded by Singapore Ministry of Education (MOE) Tier 1 Academic Research Fund No. MSS23C013.

## References

Antoine Chaffin, Vincent Claveau, and Ewa Kijak. 2022. PPL-MCTS: Constrained textual generation through discriminator-guided MCTS decoding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2953–2967.

Huy Dao, Yang Deng, Dung D. Le, and Lizi Liao. 2024. 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.

Yang Deng, Wenxuan Zhang, Wai Lam, See-Kiong Ng, and Tat-Seng Chua. 2023a. [Plug-and-play policy planner for large language model powered dialogue agents](#).

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

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

Shirley Anugrah Hayati, Dongyeop Kang, Qingxi-aoyang Zhu, Weiyan Shi, and Zhou Yu. 2020. IN-SPIRED: Toward sociable recommendation dialog systems. In *EMNLP*, pages 8142–8152.

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

Youngsoo Jang, Jongmin Lee, and Kee-Eung Kim. 2020. Bayes-adaptive monte-carlo planning and learning for goal-oriented dialogues. In *AAAI Conference on Artificial Intelligence*.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *NeurIPS*.

Shuokai Li, Ruobing Xie, Yongchun Zhu, Xiang Ao, Fuzhen Zhuang, and Qing He. 2022. User-centric conversational recommendation with multi-aspect user modeling. In *SIGIR*.

Jinggui Liang and Lizi Liao. 2023. ClusterPrompt: Cluster semantic enhanced prompt learning for new intent discovery. In *Findings of the Association for*

- Computational Linguistics: EMNLP 2023*, pages 10468–10481. Association for Computational Linguistics.
- Jinggui Liang, Lizi Liao, Hao Fei, and Jing Jiang. 2024a. Synergizing large language models and pre-trained smaller models for conversational intent discovery. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14133–14147. Association for Computational Linguistics.
- Jinggui Liang, Lizi Liao, Hao Fei, Bobo Li, and Jing Jiang. 2024b. Actively learn from LLMs with uncertainty propagation for generalized category discovery. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7845–7858. Association for Computational Linguistics.
- Lizi Liao, Yunshan Ma, Xiangnan He, Richang Hong, and Tat-seng Chua. 2018. Knowledge-aware multimodal dialogue systems. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 801–809.
- Lizi Liao, Grace Hui Yang, and Chirag Shah. 2023. Proactive conversational agents in the post-chatgpt world. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 3452–3455.
- Dongding Lin, Jian Wang, and Wenjie Li. 2023. Cola: Improving conversational recommender systems by collaborative augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4462–4470.
- 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. 2020. Towards conversational recommendation over multi-type dialogs. In *ACL*, pages 1036–1049.
- Wenchang Ma, Ryuichi Takanobu, and Minlie Huang. 2021. CR-walker: Tree-structured graph reasoning and dialog acts for conversational recommendation. In *EMNLP*, pages 1839–1851.
- Mary McHugh. 2012. Interrater reliability: The kappa statistic. *Biochemia medica : časopis Hrvatskoga društva medicinskih biokemičara / HDMB*, 22:276–82.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Jian Wang, Yi Cheng, Dongding Lin, Chak Leong, and Wenjie Li. 2023a. 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.
- Jian Wang, Dongding Lin, and Wenjie Li. 2023b. Dialogue planning via brownian bridge stochastic process for goal-directed proactive dialogue. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 370–387.
- 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.
- Yuxia Wu, Lizi Liao, Gangyi Zhang, Wenqiang Lei, Guoshuai Zhao, Xueming Qian, and Tat-Seng Chua. 2022. State graph reasoning for multimodal conversational recommendation. *IEEE Transactions on Multimedia*.
- Chenchen Ye, Lizi Liao, Suyu Liu, and Tat-Seng Chua. 2022. Reflecting on experiences for response generation. In *ACM Multimedia*, pages 5265–5273.
- 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.
- Shun Zhang, Zhenfang Chen, Yikang Shen, Mingyu Ding, Joshua B. Tenenbaum, and Chuang Gan. 2023. Planning with large language models for code generation. In *The Eleventh International Conference on Learning Representations*.
- Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2023. Language agent tree search unifies reasoning acting and planning in language models.
- Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020. Towards topic-guided conversational recommender system. In *COLING*, pages 4128–4139.

## A Appendix

### A.1 The Completed T-EPL Algorithm

In this section, we introduce details regarding the proposed algorithm T-EPL. Similar to MCTS-based algorithms (Yu et al., 2023; Jang et al., 2020), T-EPL employ a 4-stages process including: **Selection**, **Expansion**, **Estimation** and **Backpropagation**.

**Selection.** Starting from the root  $s \leftarrow s_t$  (where  $s_t$  is the current dialogue state), unless we reach a leaf node, we employ Upper Confidence bounds applied to Tree (UCT) (Yu et al., 2023) to choose a node for tree traversal. Formally, at each step, we travel the tree by using the following recursive formulation:

$$s \leftarrow \arg \max_{s' \in \text{Children}(s)} \left[ V(s') + c \pi_p(a_{s'}|s) \sqrt{\frac{2N(s)}{N(s')}} \right],$$

where  $\pi_p(a_{s'}|s; \theta)$  is a backbone policy, offering prior assessment over actions during tree-search. For T-EPL, we leverage RTCP ( $\alpha = 0$ ) (Dao et al., 2023) as our backbone policy. Furthermore,  $N(s), N(s')$  are the numbers that the parent state  $s$  and the child node  $s'$  are visited, respectively.  $c$  is a hyper-parameter to balance between exploitation and exploration aspects.

**Expansion:** After reaching a leaf node  $s$ , we utilize the backbone policy  $\pi_p(a|s; \theta)$  to sample a set of potential actions  $\hat{A}_{t+1}$ . We employ an action  $\hat{a}_{t+1} \in \hat{A}_{t+1}$  and move to the new state  $\hat{s}_{t+1}$ . If this newly constructed state is not in the current tree, we create a new node for this state.

**Estimation:** Given the state of the newly constructed node  $\hat{s}_{t+1}$  and the predefined target item  $v$ , we compute the experiential target-driven scoring function  $F_{\mathcal{M}}(\hat{s}_{t+1}, v)$  using E.q. 1.

**Back Propagation:** We then update the statistics of nodes  $s$  along the traveled path using the recursive formulation detailed in Algorithm 1.

After running the tree search for a number of simulation steps  $n$ , we choose the action  $\hat{a}_{t+1}$  of the root node  $s_t$  that leads to the highest quantity  $R(s, \hat{a}_{t+1}, \hat{s}_{t+1}) + V(\hat{s}_{t+1})$  and use it to generate a response.

### A.2 Discussions

**T-EPL and Existing MCTS-based Dialogue Policies.** First, our proposed T-EPL algorithm addresses the issue of sampling inefficiency of vanilla MCTS-based algorithms (Chaffin et al., 2022; Hao

et al., 2023; Zhou et al., 2023; Zhang et al., 2023). Specifically, such algorithms aim to estimate the potential value of the current state by using Monte-Carlo simulations, sampling completed trajectories whose outcomes will be viable at the terminated states. To accurately estimate the state’s value, it is required to run these rollout steps multiple times and compute the averaged result. However, in the worst-case scenario, the process continues until the generated conversations reach a predefined maximum length (denoted by  $H$ ). In the context of target-driven dialogues, this characteristic causes vanilla MCTS-based algorithms to exhibit a computational cost, quadratically scaling with the conversation horizon (i.e.,  $\mathcal{O}(H^2)$ ). Such an extremely high complexity hinders their applications in real-world scenarios.

To address this limitation, some existing approaches propose to utilize deep neural networks (Wang et al., 2020) or LLMs (Yu et al., 2023) for state value evaluation, effectively bypassing the costly rollout processes. However, these methods require pre-training the value functions on a limited amount of simulated interactions that may not accurately reflect real-world human conversations. Moreover, such pre-trained neural networks are fixed after the training stage, limiting their adaptability to newly encountered interactions during inference. Additionally, despite showing promising performance in various tasks, such as conversational intent discovery (Liang and Liao, 2023; Liang et al., 2024a,b), seamlessly invoking LLMs at inference time for state value estimation offers a potential solution, the frequent API calls involved can introduce significant computational overhead, making it impractical for real-world applications.

Hence, we propose T-EPL, a novel dialogue policy method that also utilizes the MCTS algorithm. In contrast to existing memory-less approaches, we employ a memory structure  $\mathcal{M}$ , storing past experienced states and their assessment values. Given the current state, T-EPL attempts to retrieve similar experienced dialogue stages and aggregate their assessment values. This approach not only allows us to bypass the costly rollout steps but also alleviates the need for either pre-trained neural networks or LLMs for state value estimations. Last but not least, while almost existing methods are offline (i.e. their policies remain static during inference), T-EPL still refines its policy at testing time by dynamically updating the memory structure. This characteristic distinguishes T-EPL from existing methods since

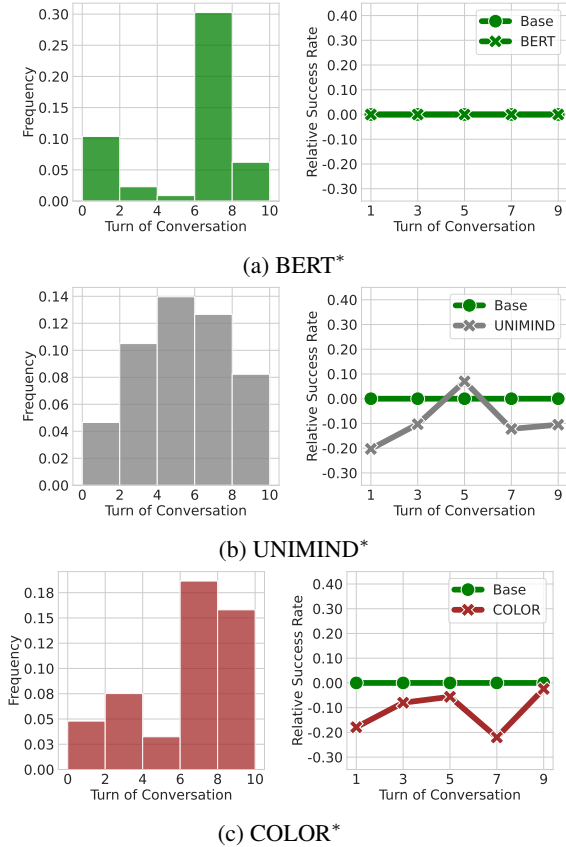


Figure 5: Frequencies of target items (left) and relative success rate (right) w.r.t conversation turns of BERT, UNIMIND, and COLOR models.

the T-EPL algorithm can be seen as an online dialogue policy.

**Reflecting on Experience.** Inspired by the self-reflection process in human psychology, we propose Experiential Policy Learning (EPL) for target-driven dialogues. Specifically, upon encountering a decision-making situation, people may attempt to recall similar experiences from their memory. By considering the outcomes of different actions taken in those past circumstances, they could choose the course of action that appears most beneficial. In a similar vein, our EPL method leverages analogous past interactions to estimate the potential values of different actions. By retrieving and aggregating past experiences, the algorithm can select the action most likely to achieve optimal results.

**Objective and Subjective Evaluation.** Prior works (Wang et al., 2022; Deng et al., 2023b; Wang et al., 2023b; Dao et al., 2023) on target-driven recommendation dialogues often evaluate the model performance by utilizing objective metrics (e.g. determining if the generated responses contain the target item). As shown in Section 5.2, this ap-

proach might overestimate the true effectiveness of target-driven recommendation systems. For example, suppose we utilize a rule-based policy that simply outputs the target item at every turn. In such a circumstance, the objective SR would be 100%. However, such a dialogue strategy is undesired as it might frustrate users and negatively impact their experience. Recognizing this limitation, we employ both objective and subjective SR metrics. While acquiring real-world human judgments can be expensive, we leverage LLMs to obtain user-simulated judgments. Additionally, unlike existing works (Wang et al., 2022; Deng et al., 2023b; Dao et al., 2023; Wang et al., 2023b), which focus on single-turn interactions, we evaluate model performance through multi-turn conversations, which are more practical and related to real-world applications.

Domain	DuRecDial 2.0	INSPIRED
Movie	190/121/161	368/42/55
Music	139/109/120	-
Food	48/13/30	-
Point-of-interest (POI)	96/42/65	-

Table 5: The detailed statistics of target items regarding different domains in DuRecDial 2.0 and INSPIRED datasets.

### A.3 Additional Statistics regarding Datasets

In this work, we conduct experiments on two published benchmark datasets, namely **DuRecDial 2.0** (Liu et al., 2021) and **INSPIRED** (Hayati et al., 2020). Besides reported statistics in Table 1, in Table 5, we further show the number of target items across various domains within the training, validation, and test sets. For DuRecDial 2.0, after the data pre-processing, we identify four domains: **Movie**, **Music**, **Food**, and **POI**. Additionally, the data exhibits a bias toward Music and Movie recommendations. This suggests that a scarcity of training data for POI and Food domains may present a greater challenge in learning effective dialogue strategies for these domains, which is further investigated in Section A.7. For the INSPIRED dataset, movie recommendations comprise the sole scenario.

In Table 6, we show detailed statistics of the dialogue strategies employed within the DuRecDial 2.0 and INSPIRED datasets. The table details the number of distinct dialogue strategies identified in each dataset. Notably, there are 13 and 14 different dialogue strategies for DuRecDial 2.0 and

INSPIRED, respectively. Furthermore, for DuRecDial 2.0, we experience that the amounts of dialogue strategies associated with the Music and Movie domains are greater compared to these two other ones. We recommend interested readers to check out their corresponding papers for more details.

DuRecDial 2.0		INSPIRED	
Strategy	Amount	Strategy	Amount
Greetings	4,948	Opinion inquiry	1,258
Ask about weather	4,393	Self modeling	235
Play music	10,034	Personal opinion	1,388
Q/A	6,072	Credibility	1,563
Music on demand	1,692	Encouragement	1,146
Movie recommendation	1,4882	Similarity	539
Chat about stars	16,276	Rephrase preference	103
Say goodbye	12,819	Preference confirmation	436
Music recommendation	13,170	Acknowledgment	814
Ask about date	2,401	Personal experience	304
Ask questions	2,100	Experience inquiry	880
POI recommendation	5,451	Offer help	449
Food recommendation	4,465	Transparency	120
		No strategy	1,423

Table 6: The detailed statistics regarding the numbers of dialogue strategies in the DuRecDial and INSPIRED datasets.

#### A.4 Recommendation Strategies of Different Dialogue Policy Methods

In this work, we compare our proposed T-EPL algorithm to various baseline methods. Specifically, these methods can be categorized into two main approaches: *predictive policies* and *generative policies*. Predictive policies, such as BERT and RTCP, aim to predict a probability distribution across dialogue actions. In contrast, generative policies, including UNIMIND, TCP, and COLOR, directly generate dialogue strategies using text generation models. As discussed in Section 5.5, we experience that two primary recommendation strategies emerge across different dialogue policies: *early recommendations* and *late recommendations*. In Figure 5, we further illustrate the frequency of target item recommendations of BERT, UNIMIND, and COLOR methods. Our findings reveal another interesting trend: predictive policies (BERT and RTCP) tend to favor both early and late recommendations, whereas generative policies often prioritize late suggestions. This might explain UNIMIND and COLOR’s lower success rates compared to BERT in early conversation turns since these generative policies simply generate fewer recommendations. Another potential hypothesis is that different domains necessitate distinct dialogue strategies for making recommendations, and there is a significant

performance gap between predictive and generative models across different recommendation domains. Section A.7 will provide detailed analyses regarding this hypothesis.

#### A.5 Performance Comparison w.r.t Different Conversation Turns

In Figure 6, we show the performance comparison of our T-EPL algorithm and several baseline methods, including TCP, UNIMIND, COLOR, and, RTCP, across various conversation turns. In particular, we evaluate the relative success rate of each model against the standard BERT baseline. Our findings demonstrate that T-EPL consistently outperforms all baseline methods across different conversation turns. Notably, in both DuRecDial 2,0 and INSPIRED datasets, we experience that the performance gap is more significant in earlier rounds. This shows that T-EPL possesses a superior capability to manage early recommendations compared to alternative dialogue policy methods. Additionally, on the INSPIRED dataset, encompassing conversations with longer horizons, T-EPL still exhibits its strength in later conversation turns. This observation also implies that our algorithm retains a robust long-range planning capability.

Figure 7 further investigates the performance comparison between T-EPL and several baseline methods, including vanilla MCTS, GDP-Zero, and PDPDP, across different conversation turns. Our proposed algorithm consistently outperforms vanilla MCTS across different conversation stages. In comparison with the strong baseline GDP-Zero, T-EPL achieves competitive results, particularly in later conversation turns. While GDP-Zero appears to perform better in the initial turn, this might be attributed to its tendency to solely focus on early recommendations. In some circumstances, this behavior might be undesired as consistently recommending the target item in the early rounds might hurt user experience and fail to establish user engagement on the targeted item. Furthermore, despite its competitive performance, GDP-Zero incurs significantly higher computational costs, as will be detailed in Section A.8.

#### A.6 Statistics of Different Dialogue Strategies w.r.t Conversation Turns

In Figure 8, we show the frequency of dialogue actions employed by different dialogue policies, including T-EPL, RTCP, and TCP, across different conversation turns. Our findings reveal that in the

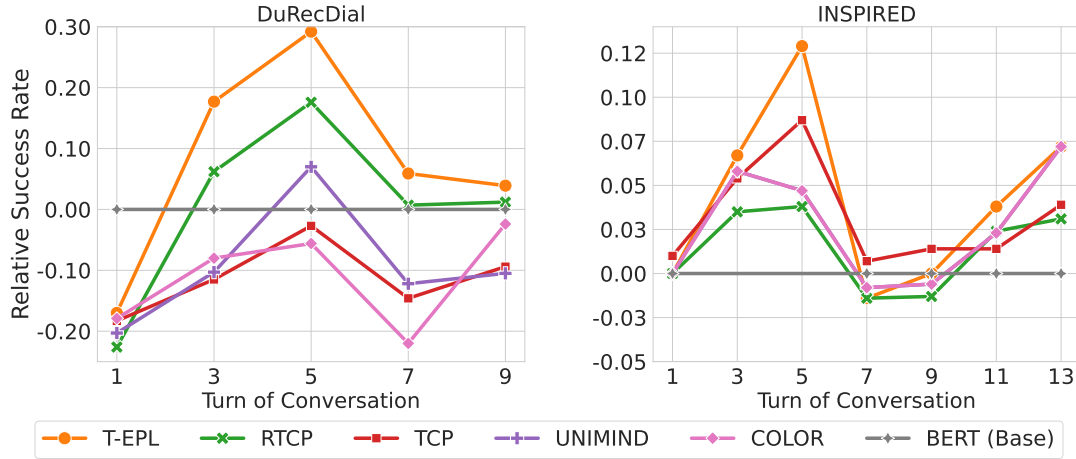


Figure 6: Comparison of relative success rate against the standard baseline (i.e. BERT) at different conversation turns. We utilize 10 and 14 conversation turns for DuRecDial and INSPIRED, respectively.

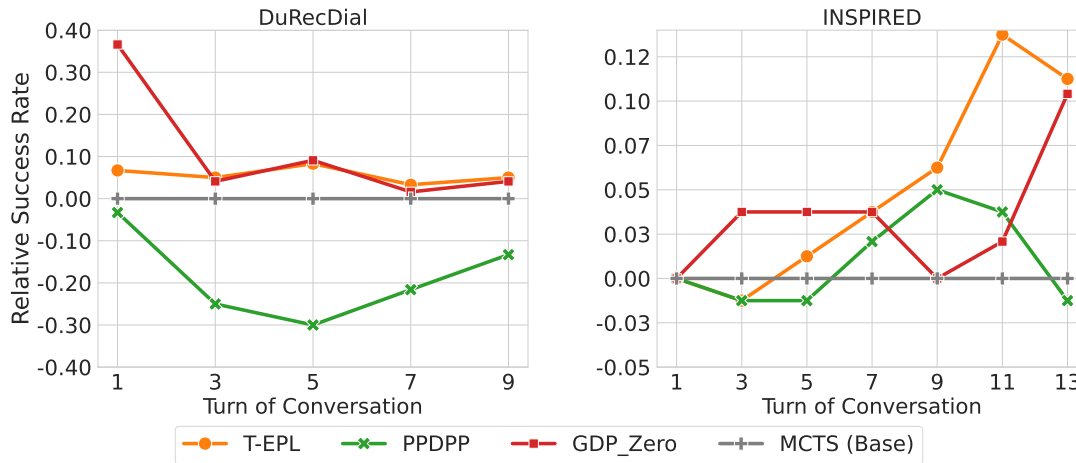


Figure 7: Comparison of relative success rate against the standard baseline (i.e. MCTS) at different conversation turns. We utilize 10 and 14 conversation turns for DuRecDial 2.0 and INSPIRED, respectively.

early stages of conversations (turns 1 to 4), these models prioritize establishing rapport with users rather than immediately offering recommendations. This initial interaction phase is crucial for fostering positive user experiences before introducing recommendations. Furthermore, between the 5th and 6th turn, we observe some changes in their dialogue policies. Particularly, the three considered policies initially provide recommendations for some specific domains, such as Music, Food, and Movie. Notably, RTCP and T-EPL exhibit a higher frequency of recommendations during this period. This aligns with our findings presented in Section 5.5, which demonstrates that these models tend to offer both early and late recommendations. Moreover, the reported statistics also reveal that the recommendation strategies might vary across different domains. Specifically, we experience a tendency to

favor music and food recommendations in the early rounds (i.e. 3rd-4th turns and 5th-6th turns), while movie and point-of-interest (POI) suggestions are preferred in later turns (i.e. 7th-8th and 9th-10th turns). This observation indicates that optimal dialogue strategies for maximizing user benefit may vary depending on the domain under consideration.

### A.7 Performance Comparison w.r.t Different Recommendation Domains

In target-driven recommendation dialogues, it is desired to investigate the performance of dialogue policy methods across diverse domains. As shown in the previous section, optimal dialogue strategies might vary with different domains of interest. Hence, in Table 8, we show the performance of our T-EPL algorithm and other baseline models on the DuRecDial 2.0 dataset across four domains

(i.e. Movie, POI, Food, and Music). Particularly, the reported results are consistently lower for POI and Food domains, compared to Movie and Music ones. This suggests that achieving successful POI and Food recommendations is more challenging. A potential explanation lies in the limited amount of training data for these two domains, which may hinder the model’s ability to learn effective dialogue policies. Furthermore, the proposed T-EPL algorithm significantly outperforms all baseline methods across 3 out of 4 domains of consideration. This demonstrates the superiority of our proposed T-EPL algorithm, compared to other existing baseline methods. Notably, this success highlights the generalizability of T-EPL, suggesting its capacity to enhance target-driven dialogue planning irrespective of the domain under consideration.

Model	Inference Time (s)	
	DuRecDial 2.0	INSPIRED
BERT (Devlin et al., 2019)	<b>6.01</b>	<b>6.62</b>
UNIMIND (Deng et al., 2023b)	7.54	9.21
TCP (Wang et al., 2022)	13.60	34.34
COLOR (Wang et al., 2023b)	10.81	26.07
RTCP (Dao et al., 2023)	7.53	8.69
MCTS	105.29	232.04
GDP-Zero (Yu et al., 2023)	90.62	148.70
PPDPP (Deng et al., 2023a)	7.59	9.49
T-EPL (ours)	50.71	84.43

Table 7: Inference time on one target item of different dialogue policy methods. In particular, the maximal number of conversation turns are 10 and 14 for DuRecDial 2.0 and INSPIRED, respectively.

### A.8 Detailed Analyses regarding Model Complexity and Computational Time

First, we provide detailed analyses regarding the API call complexities of T-EPL and other baseline models at inference time. Suppose  $T$ ,  $H$ , and  $K$  are the number of target items, the conversation horizon, and the number of simulation steps used in MCTS-based algorithms, respectively. It is straightforward to show that the offline policies (i.e. BERT, UNIMIND, COLOR, and RTCP) require  $T \times H$  API calls to evaluate the whole item set. Therefore, the upper bound of their computational complexities is  $\mathcal{O}(TH)$ . For vanilla MCTS, due to the need for rollout steps, its complexity scales quadratically with the conversation horizon  $H$ . This makes the overall complexity of MCTS  $\mathcal{O}(TKH^2)$ . Regarding GDP-Zero, it utilizes LLMs for state value estimation. In particular, at each simulation step, GDP-Zero needs to call LLMs for  $N \geq 1$  times

to compute the state value. Following official implementations of GDP-Zero, we utilize  $N = H$ , making its complexity  $\mathcal{O}(TKH^2)$ . Finally, our T-EPL policy exhibits a complexity of  $\mathcal{O}(TKH)$ . This is reasonable since T-EPL estimates the state’s potential score using past experienced interactions. Therefore, it does not require costly rollout steps and LLM-based evaluations during inference.

To approximately validate these derived complexities, in Table 7, we present the inference time on one target item of considered models. In particular, the observed inference times align with the corresponding API call complexities, where a greater number of API calls correspond to a higher inference time. Notably, our T-EPL algorithm demonstrates a significant advantage compared to other MCTS-based methods, including vanilla MCTS and GDP-Zero.

### A.9 Additional Details of Baseline Methods

In this work, we compare our T-EPL algorithm against several representative baselines, including:

- **BERT**<sup>4</sup> (Devlin et al., 2019) is a general pre-trained language model based on Transformer Encoder. Specifically, we utilize BERT to predict the next dialogue strategy.
- **TCP**<sup>5</sup> (Wang et al., 2022) is an early target-driven recommender system. In particular, it leverages a text generation model to generate a sequence of actions, starting from the target action to the current turn.
- **UNIMIND**<sup>6</sup> (Deng et al., 2023b) is a goal-aware conversational recommender system that utilizes a multi-task learning paradigm and a prompt-based learning to unify sub-tasks of multi-goal CRS setting.
- **COLOR**<sup>7</sup> (Wang et al., 2023b) is a recent target-driven dialogue system aiming to learn latent traditions within the dialogues via the Brownian-motion bridge.
- **RTCP**<sup>8</sup> (Dao et al., 2023) is the state-of-the-art target-driven recommendation model that aims to direct the conversation with a short-term and a long-term planning module. Additionally, this model balances these two aforementioned modules with a strategic balancing

<sup>4</sup><https://huggingface.co/google-bert/bert-base-cased>

<sup>5</sup><https://github.com/iwangjian/Plan4RecDial/tree/main>

<sup>6</sup><https://github.com/dengyang17/UniMIND>

<sup>7</sup><https://github.com/iwangjian/Color4Dial>

<sup>8</sup><https://github.com/rtcpproject/RTCP>

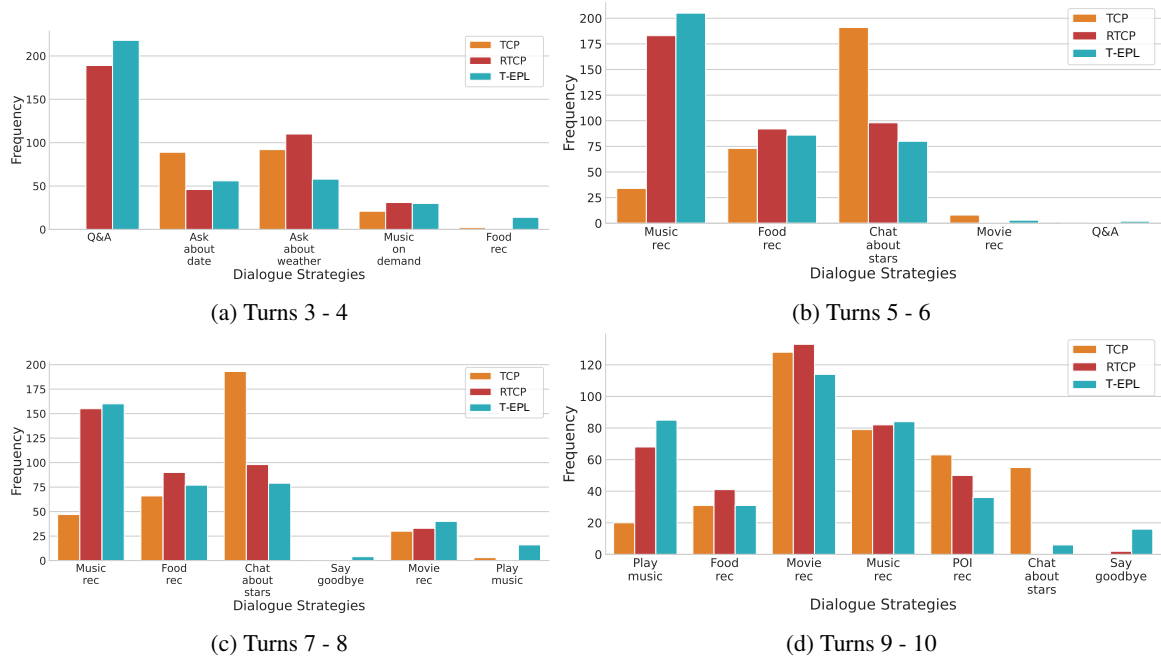


Figure 8: Frequencies of predicted dialogue strategies w.r.t different conversation turns of TCP, RTCP, and T-EPL. In particular, we consider the DuRecDial 2.0 dataset and report the statistics accordingly.

mechanism.

- **GDP-Zero**<sup>9</sup> (Yu et al., 2023) is a recent target-driven dialogue system. Specifically, it utilizes open-loop MCTS to perform look-ahead planning, where state values are produced by prompting an LLM model.
- **PPDPP**<sup>10</sup> (Deng et al., 2023b) is a recent target-driven dialogue system. In particular, this model leverages some background datasets to finetune 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.

In this work, we leverage **the published source codes** of baseline methods to conduct experiments. Moreover, for the ablation study, we compared our T-EPL with other variants described as follows:

- *w/o Len*: The variant without the length constraint in E.q 2 (i.e  $\lambda = 0$ ).
- *w/o Exp*: The variant without the exponential target-driven scoring function (i.e  $F_{\mathcal{M}}(s, v) = 0$ ).

## A.10 Implementation Details

We implement our T-EPL algorithm using the PyTorch framework<sup>11</sup>. Experiments were conducted on a single NVIDIA A100 40GB GPU card. Moreover, the performance evaluation protocols are detailed as follows:

- **Full Test Set** (\*): For this circumstance, we use the whole test set (376 and 55 target items for DuRecDial 2.0 and INSPIRED, respectively) and execute each model twice with varying random seeds. The reported results represent the average of these runs.
- **Comparison against MCTS, GDP-Zero, and PPDPP** ( $\mp$ ): Due to the high computational demands of these baselines, we follow the approach outlined by Yu et al. (2023). This involved randomly selecting 20 target items (i.e. 100 and 140 interaction rounds in total for DuRecDial 2.0 and INSPIRED, respectively) from the test set for automatic evaluation. Each model was then run three times with varied random seeds, and the final results were obtained by averaging these runs.

For T-EPL’s implementation, we utilize the RTCP policy ( $\alpha = 0$ ) (Dao et al., 2023) as the backbone policy. Specifically, the policy was first fine-tuned on the training data from the corresponding background dataset (i.e. DuRecDial 2.0 or INSPIRED)

<sup>9</sup><https://github.com/jasonyux/GDPZero>

<sup>10</sup><https://github.com/dengyang17/PPDPP>

<sup>11</sup><https://pytorch.org/>



Model	Movie		Music		POI		Food	
	Subj <sub>sr</sub>	Avg. T(↓)	Subj <sub>sr</sub>	Avg. T(↓)	Subj <sub>sr</sub>	Avg. T(↓)	Subj <sub>sr</sub>	Avg. T(↓)
BERT*	0.869	6.782	0.833	4.333	0.723	6.938	0.533	6.600
UNIMIND*	0.788	6.503	0.858	4.825	0.261	7.769	0.500	7.566
TCP*	0.832	6.881	0.791	5.975	0.569	7.000	0.200	8.600
COLOR*	0.782	7.391	0.800	6.700	0.584	8.153	0.533	5.553
RTCP*	<b>0.925</b>	6.204	0.875	4.375	0.738	7.107	0.333	7.773
T-EPL* (ours)	0.851	<b>5.708</b>	<b>0.891</b>	<b>3.891</b>	<b>0.831</b>	<b>4.892</b>	<b>0.667</b>	<b>5.133</b>

Table 8: Performance comparison w.r.t different domains. Specifically, we report the results on four domains: Movie, Music, Point-of-Interest (POI), and Music Recommendation. The final results are reported on the whole test set (t-test,  $p < 0.05$ ).

to predict the next action. This fine-tuning process involved training the policy model for 5 epochs with learning rates of  $5e-5$  and  $1e-5$  for DuRecDial 2.0 and INSPIRED datasets, respectively. Following fine-tuning, the pre-trained policy was employed to construct the memory component (denoted by  $\mathcal{M}$ ), as described in Section 4.2. The dimension of the hidden vectors within this memory was set to 768. Moreover, we set the number of simulation steps ( $n$ ) and the size of the memory buffer  $k$  to 5 and 20, respectively. Finally, we regard a generated conversation as a successful case if its target-driven assessment score surpasses a threshold of  $\epsilon = 1$ .

### A.11 Response Generation Model

This study primarily focuses on evaluating the planning capabilities of our proposed T-EPL algorithm in comparison to existing dialogue policy methods. To isolate the final performance from other aspects, such as the response generation model as well as to ensure a fair comparison, we employ the same text generation model for all dialogue policy methods under investigation. Specifically, a BART-based model with 114M parameters is utilized for text generation across all considered dialogue policies. The input provided to the text generation model is constructed by combining three elements, namely predicted dialogue strategies and topics from the dialogue policies and the current dialogue history. For model training, we set the number of training epochs and learning rate to 5 epochs and  $5e-5$  for both DuRecDial 2.0 and INSPIRED datasets.

### A.12 User Response Simulation

In this section, we describe the role-playing prompt employed to instruct LLMs in simulating user responses. In particular, we utilize a 1-shot prompting scheme, where the input of the LLM is constructed by using the current dialogue history, the

newly generated system response, and one demonstrative conversation. The prompt utilized for the user simulator is detailed in Table 9. In particular, we prompt the LLM with temperature  $\rho = 0$ . To avoid bias during the dialogues, the targeted item should never be mentioned in the designed prompt for user response generation.

### A.13 LLM-based Target-driven Assessment

In target-driven recommendation dialogues, it is crucial to determine if the user is happy and willing to accept the target item recommended by the system. This section outlines the instruction prompt employed to compute an LLM-based target-driven assessment. The specific details of this prompt are presented in Table 10. In particular, we prompt the LLM with temperature  $\rho = 1.1$  for 5 times and compute the averaged score.

### A.14 Qualitative Analyses

In Section 5.5, we hypothesize that there are two potential dialogue strategies that T-EPL can leverage to introduce target items to users. To demonstrate this hypothesis, Table 11 presents example conversations generated by our T-EPL algorithm. In the first conversation, T-EPL attempts to introduce the target item, **[Forever Young]**, early in the interaction sequence, specifically during the second turn. Subsequently, T-EPL seeks to further persuade the user by providing additional details about the recommended item, such as mentioning its actor, **Xiaoming Huang**. Conversely, the second conversation example showcases a different approach. Here, T-EPL initially presents relevant information, such as mentioning **Hsu Chi** and **Bo Huang**. Only after the user expresses interest does T-EPL introduce the target item, **The Island**. These examples demonstrate T-EPL’s capability to employ diverse dialogue strategies when making recommendations.

	Role	Utterance
<b>[Target Goal]: [Movie recommendation]</b>		
<b>1-shot example</b>	system	This is an example of a <b>Movie recommendation</b> conversation between the user (you) and the system.
	assistant	[1] How tall is Xiaoming Huang?
	user	He's 180cm tall.
	assistant	I know. Thank you. You know so much.
	user	[2] I also know that he has won Best Actor of the Year at the 10th Chinese American Film Festival.
	assistant	Yeah, he's very good at acting. I like him very much.
	user	As an actor, he has been improving his acting skills. He is really a good actor. I'm not surprised you like him.
	assistant	I'm glad you can say that, and I will always support him.
	user	[3] Then you can go to see his movie Sacrifice. It's said to be a good movie.
	assistant	Who directed the film?
	user	Kaige Chen and Chongjie Ma.
	assistant	Wow, it's amazing. I must watch it.
	user	[4] All right. I'm going for lunch. Talk to you next time. Bye!
assistant	Bye!	
<b>Dialog History</b>	system	Now enter the role-playing mode. In the following conversation, you will play as a user. You are the user who is looking for a <b>Movie recommendation</b> . Please reply with only one short and succinct sentence.
	user	Hello ! How do I help you?
	assistant	Hi! I'm looking for a movie recommendation. Is there any suggestion?
	user	It's Bingbing Fan's New Shaolin Temple.
	assistant	Thank you for the recommendation!
	user	You're welcome. Bingbing Fan has won the Hundred Flowers Awards for Best Actress.
<b>User Response</b>	assistant	That's impressive! I'll definitely check out the movie.

Table 9: Detailed instruction prompt for user simulator for a movie recommendation conversation. "Role" is identical to the role in the ChatGPT prompting scheme. For user response generation, we exchange the roles of the system and the user in the instruction prompt. It is worth noticing that the target item of the current conversation should never be indicated directly in the instruction prompt.

### A.15 Instructions for Human Evaluation

Given generated dialogues by our T-EPL and other baseline methods, we ask the annotators to compare those dialogues in both **Satisfaction** and **Coherency** aspects, defined via following questions:

- **Satisfaction:** Given the target item, which dialogue offers more convincing justifications to accept the target item ?
- **Coherency:** Given the target item, which dialogues offer more reasonable topical transitions towards the target item ?

	Role	Utterance
<b>[Target Item]: [New Shaolin Temple]</b>		
<b>Conversation</b>	system	The following is a new conversation between a recommender and an user.
	assistant	Hello ! How do I help you?
	user	Hi! I'm looking for a movie recommendation. Is there any suggestion?
	assistant	It's Bingbing Fan's New Shaolin Temple.
	user	Thank you for the recommendation!
	assistant	You're welcome. Bingbing Fan has won the Hundred Flowers Awards for Best Actress.
	user	That's impressive! I'll definitely check out the movie.
	assistant	You have a good taste. Bingbing Fan has also won the Hundred Flowers Awards for Best Actress.
	user	Thank you! I appreciate the recommendation.
	assistant	You're welcome. Bingbing Fan's New Shaolin Temple is also very good. You can watch it.
	user	I will definitely add it to my watchlist. Thank you!
	assistant	You're welcome. I'm sure you'll like it.
	user	I look forward to watching it!
	system	Based on the given conversation, you need to infer the attitude of the user towards the target item: [New Shaolin Temple]. You need to infer if the user is happy and willing to accept the target item: [New Shaolin Temple]. If the user is happy, you need to generate the word: [accept]. If the user is confused or not willing to accept the item: [New Shaolin Temple], you need to generate the word: [reject].
<b>Results</b>	assistant	[accept, accept, accept, accept, accept]

Table 10: Detailed instruction prompt for LLM-base target-driven assessment for a movie recommendation conversation. "Role" is identical to the role in the ChatGPT prompting scheme. After getting the results from ChatGPT, we convert textual assessments to corresponding scalar values (i.e. `accept` = 1, `reject` = -1) and compute the averaged score.

	Role	Utterance
<b>[Target Item]: [Forever Young]</b>		
<b>Example 1</b>	T-EPL	Hello! How can I help you?
	User	I'm looking for a good movie to watch, any recommendations?
	T-EPL	It's Xiaoming Huang's <b>Forever Young</b>
	User	Thank you for the recommendation!
	T-EPL	You're welcome. <b>Xiaoming Huang</b> is the Most Popular Actor in the 17th Beijing College Student Film Festival
	User	That's impressive!
	T-EPL	Yes. He also won the Most Anticipated Actor of the Chinese Film Media Award.
	User	He must be very talented
	T-EPL	Yes, his movie <b>Forever Young</b> is very good. You can watch it.
	User	I'll definitely check it out, thanks!
	T-EPL	You're welcome. I'm sure you'll like it.
	User	I appreciate the recommendation.
<b>[Target Item]: [The Island]</b>		
<b>Example 2</b>	T-EPL	Hello ! How do I help you ?
	User	I'm looking for a good action movie to watch, any recommendations?
	T-EPL	<b>Hsu Chi</b> is a popular movie star. Do you like her ?
	User	Yes, I like her a lot !
	T-EPL	Do you like <b>Bo Huang</b> ? He won the Golden Horse Award for Best Actor.
	User	Yes, I think <b>Bo Huang</b> is a talented actor!
	T-EPL	Yes, he has also won the Asian Film Awards for Best Actress.
	User	<b>Bo Huang</b> is actually a male actor, not a female actress.
	T-EPL	Yes, I'd like to recommend his movie <b>The Island</b> to you. It's a good action movie.
	User	Thank you for the recommendation, I'll check out <b>The Island</b> !
	T-EPL	I'm glad you like it.
	User	Thank you!

Table 11: Two interactive conversations between our T-EPL agent and the LLM-based simulator, showing two common dialogue strategies to recommend the target items. For the first example. T-EPL aims to recommend the target item **[Forever Young]** at early rounds and offer further justifications in later turns. In contrast, in the second example, T-EPL first offers interacting information in early turns while aiming to nudge the user with the target item **[The Island]** at later turns.