Simulation-Free Hierarchical Latent Policy Planning for Proactive Dialogues

Tao He[1*], Lizi Liao[2], Yixin Cao[3], Yuanxing Liu[1], Yiheng Sun[1], Zerui Chen[1], Ming Liu[1†], Bing Qin[1]

[1]Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, Harbin, China
[2]Singapore Management University, Singapore
[3]School of Computer Science, Fudan University
{the, yxliu, mliu, zrchen, qinb}@ir.hit.edu.cn, lzliao@smu.edu.sg, caoyixin2011@gmail.com

Abstract

Recent advancements in proactive dialogues have garnered significant attention, particularly for more complex objectives (e.g. emotion support and persuasion). Unlike traditional task-oriented dialogues, proactive dialogues demand advanced policy planning and adaptability, requiring rich scenarios and comprehensive policy repositories to develop such systems. However, existing approaches tend to rely on Large Language Models (LLMs) for user simulation and online learning, leading to biases that diverge from realistic scenarios and result in suboptimal efficiency. Moreover, these methods depend on manually defined, context-independent, coarse-grained policies, which not only incur high expert costs but also raise concerns regarding their completeness. In our work, we highlight the potential for automatically discovering policies directly from raw, real-world dialogue records. To this end, we introduce a novel dialogue policy planning framework, LDPP. It fully automates the process from mining policies in dialogue records to learning policy planning. Specifically, we employ a variant of the Variational Autoencoder to discover fine-grained policies represented as latent vectors. After automatically annotating the data with these latent policy labels, we propose an Offline Hierarchical Reinforcement Learning (RL) algorithm in the latent space to develop effective policy planning capabilities. Our experiments demonstrate that LDPP outperforms existing methods on two proactive scenarios, even surpassing ChatGPT with only a 1.8-billion-parameter LLM.

Introduction

In recent years, there has been a surge of interest in dialogue tasks that require proactive engagement to achieve complex objectives, such as negotiation (He et al. 2018), persuasion (Samad et al. 2022), and emotional support (Cheng et al. 2022). Unlike traditional task-oriented dialogues (Liu et al. 2022; Hu et al. 2023; Liu et al. 2023), these tasks require agents to be more proactive and possess sophisticated dialogue strategy skills (Cheng et al. 2024). Previous research has demonstrated that even LLMs often struggle on such tasks (Yang, Li, and Quan 2021; Zhao et al. 2023; Kang et al. 2024; Song et al. 2024). LLMs are typically trained to passively follow user instructions, which leads them to align with the user's opinions and decisions, lacking the necessary proactivity (Deng et al. 2023b; He et al. 2024).

The advancement of LLMs in instruction-following and text generation capabilities has provided a foundation for exploring proactive dialogue systems, allowing a focus on high-level strategic research, i.e. dialogue policy planning (Deng et al. 2023b), which plans the next dialogue policy to guide generating appropriate responses. Some efforts have sought to directly enhance the strategic capabilities of LLMs by designing heuristic prompts or complex prompting processes (Deng et al. 2023a; Yu, Chen, and Yu 2023). However, these approaches often face limitations in performance or are criticized for high inference costs and inefficiency due to the need for continuous interactions. Other approaches aim to develop specialized policy planners to guide LLM responses strategically (Deng et al. 2023b), allowing the separation of strategy from LLM and enabling a focused effort on learning policy planning capabilities.

However, developing advanced policy planners requires rich exposure to **diverse dialogue scenarios** and access to a **comprehensive policy repository**. Previous works (Deng et al. 2023a) have used LLM like ChatGPT to simulate interactions, engaging in role-play and real-time learning. This methodology presents two critical drawbacks: first, the significant disparity between simulated and real-world interactions, as the toneless communication style of ChatGPT contrasts with the diverse and dynamic traits of actual human users; second, the reliance on continuous real-time interactions and frequent API calls for training, which introduces inefficiencies and escalates costs. Moreover, these approaches often depend on manually defined, context-independent, coarse-grained dialogue policies (Zhou et al. 2019; Liu et al. 2021a), which not only require substantial expert involvement but also raise concerns about the completeness and effectiveness of predefined policies.

In this study, we introduce a novel paradigm that shifts away from relying on predefined policy sets and online learning in simulated environments, instead directly learning policy planning from raw, unlabeled dialogue records. This paradigm effectively addresses two key challenges: 1) It allows for **discovering fine-grained policies directly from realistic dialogues**, reducing the need for expert intervention and enhancing the completeness and relevance of re-

[*]Work was done during an internship at SMU.
[†]Corresponding Author: Ming Liu.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

sulting policies. 2) By learning from real-world dialogues, it **eliminates the dependence on simulated environments**, thereby improving both efficiency and effectiveness.

To achieve this, we propose the innovative **Latent Dialogue Policy Planning (LDPP)** framework. LDPP automatically discovers policies as continuous latent vectors, expressing more semantics than predefined context-free policies, and facilitates the learning of effective planning within this latent policy space. The framework consists of three key stages: **Latent policy discovery**, **Latent policy distillation**, and **Offline Hierarchical RL enhancement**. Inspired by the Variational Autoencoder's (VAE) ability to encode inputs into an interpolable latent space (Kingma and Welling 2013), we first employ a variant of the VQ-VAE (van den Oord, Kalchbrenner, and Kavukcuoglu 2016) to automatically discover latent policies from dialogue records. These discovered latent policies are then used to label the training data. Finally, we propose an Offline Hierarchical Reinforcement Learning algorithm to both enhance the high-level policy planning and optimize response generation given latent policies at the lower token level. Since the latent policies are represented as continuous vectors rather than natural language tokens, we further introduce the P-Former module. This module functions as a trainable adapter, ensuring that LLMs can effectively understand and follow the guidance of latent policies to respond, term as the latent-policy-following ability. During inference, the policy planner first determines the appropriate latent policy based on the current dialogue state, which then directs the LLM in generating contextually relevant responses.

To verify our approach, we conducted experiments widely on ExTES (Zheng et al. 2023a), ESConv (Liu et al. 2021b) and P4G (Wang et al. 2019b). We compare our method with various baselines, demonstrating its effectiveness. Detailed analysis experiments further support the framework's validity. Our contributions are as follows:

- We introduce a novel simulation-free dialogue policy planning learning framework, automatically mining potential policies from raw dialogue records.
- We propose an offline hierarchical reinforcement learning method for optimizing proactive dialogue, improving both planning capability and latent-policy-following ability for response generation.
- Extensive experiments across three proactive dialogue benchmarks show our approach outperforms baselines, with analysis confirming its effectiveness.

Related Work

Policy Planning for LLM-powered Dialogue Agent. The advent of LLMs enables research into more complex dialogue tasks (Cheng et al. 2024) like emotion support and price negotiation. However, current studies indicate that LLMs often underperform in such tasks due to insufficient policy planning capacities (Chen et al. 2023). To improve policy planning, recent research has proposed various methods, which can be categorized into two parts: 1) *With predefined dialogue policy.* These methods need predefined dialogue policies, which can be further divided into two parts. Firstly, Deng et al. (2023a) design a prompt process requiring LLMs to select an appropriate policy before generating a response. GDP-Zero (Yu, Chen, and Yu 2023) employs Markov Monte Carlo Tree Search (Liebana et al. 2015) to identify the next strategy. However, these methods are hindered by either the fixed parameters of LLMs or their high computational costs. To overcome this, PPDPP (Deng et al. 2023b) trains a specialized policy planner via online interaction with a simulated environment. Zhang et al. (2024) increase richer user simulations to improve planning performance. DPDP (He et al. 2024) employs the Dual-process theory (Kahneman 2003) to balance the efficiency and performance. However, these methods require real-time interaction with a simulated environment, suffering from low efficiency and gaps between the realistic and simulated environment. 2) *Without predefined dialogue policy.* These approaches do not require pre-defined dialogue policies. Instead, they drive LLMs to analyze the current dialogue state and generate AI feedback, which is then used to help the LLMs to reply (Fu et al. 2023; Zhang, Naradowsky, and Miyao 2023). However, these methods often struggle to enhance the strategic reasoning capabilities of LLMs, resulting in less coherent and contextually appropriate responses, which leads to suboptimal performance.

Dialogue Generation on Latent Space. In the past years, studies have utilized latent features to control or enhance response generation (Wang et al. 2020; Cho et al. 2023; Lubis et al. 2020). Some works employ VAE (Bowman et al. 2015) variants such as CVAE (Zhao, Zhao, and Eskénazi 2017), and Discrete VAE (Bao et al. 2019) to model the semantic distribution of utterances in the latent space (Liu, Pan, and Luo 2020; Chen et al. 2022), sampling latent variables to enhance response diversity (Xiang et al. 2024). In this work, we focus on dialogue policy planning for LLM-powered proactive dialogues. We discover latent policies automatically and conduct planning within the latent space.

Preliminaries

Problem formalization. Unlike previous works that focus solely on dialogue policy planning (Deng et al. 2023b), our approach also optimizes the policy following ability for responding. To achieve this, we model the entire dialogue process using a hierarchical Markov Decision Process (MDP), inspired by recent studies (Zhou et al. 2024). At the high level, a policy-level MDP is employed to model the policy planning task, while at the low level, a token-level MDP models the autoregressive generation of responses.

The policy-level MDP is defined as $\mathcal{M}_h = \langle \mathcal{S}_h, \mathcal{A}_h, \mathcal{R}_h \rangle$. The state set $\mathcal{S}_h$ consists of the dialogue history h_t with alternating user utterances and system responses $\{u_1^{sys}, u_1^{usr}, \ldots, u_{t-1}^{sys}, u_{t-1}^{usr}\}$. The action $z_t \in \mathcal{A}_h$ refers to the dialogue policy, i.e., latent policy in this work. The reward function $\mathcal{R}_h$ evaluates each dialogue state using ChatGPT, outputting rewards r_t for each turn of dialogue. Please refer to the Evaluation Methods Section for details. Similarly, the token-level MDP is defined as $\mathcal{M}_l = \langle \mathcal{S}_l, \mathcal{A}_l, \mathcal{R}_l \rangle$, where the state set $\mathcal{S}_l = \{s_i^t = [h_t; z_t; w_{1:i-1}]\}$, with w_i representing the i-th token of the response $u_t^{sys} =$

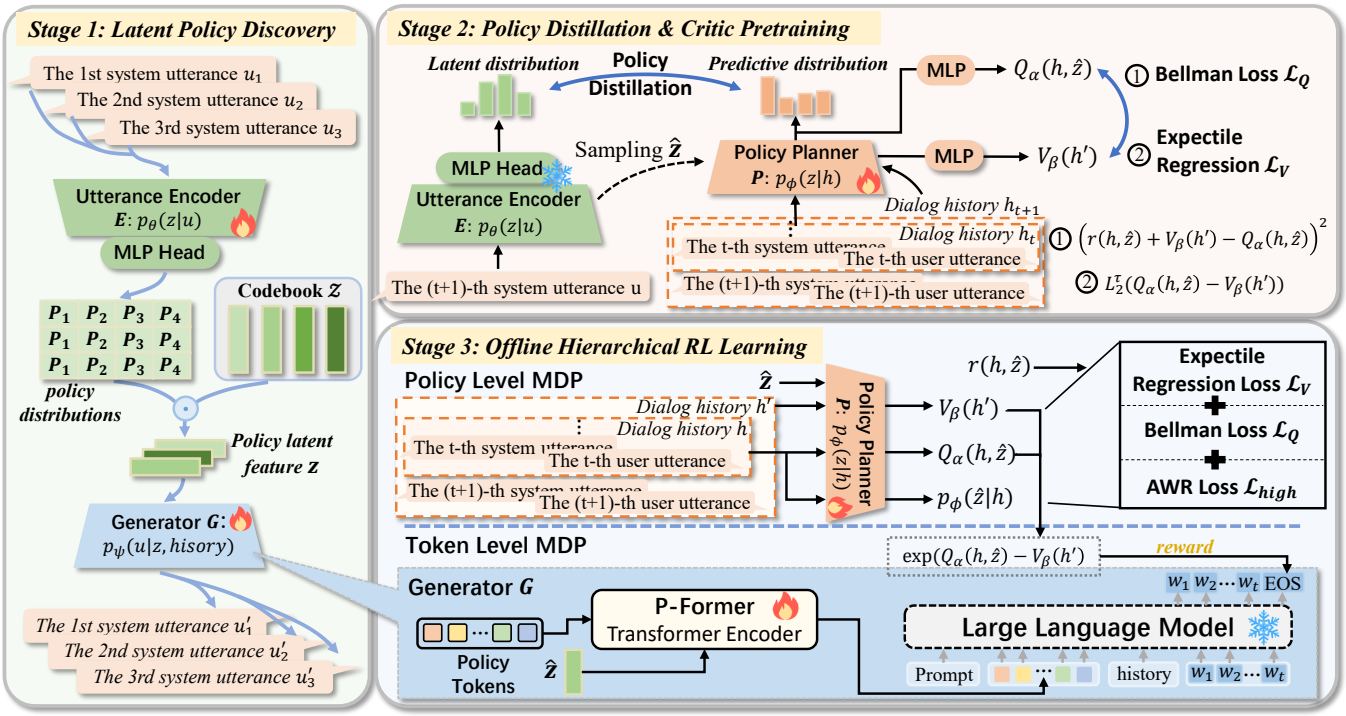


Figure 1: The training process of the LDPP framework. u and z refer to the system utterance and contained latent policy. h and h' denote t-th dialogue history h_t and $(t+1)$-th dialogue history h_{t+1}, respectively.

$[w_1, w_2, \ldots, w_n]$. The action set $\mathcal{A}_l$ is the LLM's vocabulary, and the reward function $\mathcal{R}_l$ is provided by the policy-level MDP, detailed in the 3rd Stage introduction. In the t-th dialogue turn, given the current state, i.e., dialogue history h_t, the policy planner predicts the appropriate dialogue policy z_t. Guided by the dialogue history h_t and the dialogue policy z_t, the LLM generates the response u_t^{sys}.

In our proposed offline scenario, we only access raw dialogue records $\mathcal{D}$. For RL training, we decompose $\mathcal{D}$ into tuples: $\mathcal{D} = \{(h_t, u_t^{sys}, u_t^{usr})\}$. To learn the policy-level MDP, we further use the policy-level reward r_t for each dialogue turn t to extend $\mathcal{D}$ as $\{(h_t, u_t^{sys}, u_t^{usr}, r_t)\}$.

Component Models

Before delving into the training framework, we first outline the component models. Our framework is composed of three key models: an **utterance encoder** E, a **policy planner** P, and a **generator** G. During the training phase, E learns to discover latent policies from system responses and then annotate pseudo labels (latent policies) for training set for subsequent optimizing P and G. In the inference phase, only P and G are actually employed: the planner P outputs the next-turn policy based on the dialogue history, and then the policy is fed into G to guide the response generation.

In this work, the design of the base models is not the central focus; therefore, we utilized RoBERTa-Large (Liu et al. 2019) as the base for both E and P, same as works like PPDPP (Deng et al. 2023b). E takes system responses as input and uses the output of "[CLS]" to analysis the distribution of latent policies contained in responses; P takes dialogue history as input and similarly outputs a predicted distribution of next-step policies. G is based on an LLM.

However, LLMs only accept texts, while the latent policy is a continuous vector, which obviously has a significant gap between them. Inspired by the development of Vision Large Models (Li et al. 2023), we propose to train a **P-Former** to bridge this gap. P-Former consists of L stacked transformer layers, taking T learnable policy tokens as input. These policy tokens interact with the latent policy features through a cross-attention mechanism. Ultimately, P-Former outputs T policy-related tokens. We hope these tokens align with the input space of LLM, thus LLM can understand and follows the guidance of latent policies for appropriate response generation. During training, the P-Former is optimized by the reconstruction loss of LLM. Notably, we freeze the LLM throughout. **Therefore, P-Former is also responsible for improving the latent-policy-following capacity of G.**

Optimizaing Framework

Our training framework is depicted in Figure 1. It consists of three stages with the following motivations and relationships: **Stage 1**: It focuses on automatically learning latent policies from raw dialogues. These latent policies serve as "annotations" for optimizing policy planning in Stage 3. **Stage 2**: It is used to initialize the policy planner, thereby accelerating and stabilizing the reinforcement learning process in Stage 3. **Stage 3**: Upon the preparatory work in Stage 1 and 2, this stage aims to enhance policy planning capabil-

ities at the policy level and further optimize latent-policy-following abilities for responding at the token level.

1st Stage: *Latent Policy Discovery.* We first automatically mine potential dialogue policies from raw dialogue records. The basic premise is that, given the dialogue history and the policy implied in one response, the dialogue agent should be able to reconstruct this response. To this end, we propose an adjusted VQ-VAE algorithm. We first compress the inputted utterance into latent policy and then apply it, along with dialogue history, to guide the LLM in reconstructing the utterance. If the reconstruction is good, we assume the learned latent policy is effective. For more details about VQ-VAE, please refer to the appendix.

Like VQ-VAE, We define a codebook $\mathcal{Z} = \{\mathcal{Z}_k \in \mathbb{R}^d\}_{k=1}^K$ with K policy vectors. Given a system utterance u_t^{sys}, shorted as u_t, we use the encoder E to compress it and classify it into K classes, yielding the policy distribution $p_\theta(z_t|u_t) \in \mathbb{R}^K$. Instead of performing a nearest neighbors lookup like VQ-VAE, we use $p_\theta(z_t|u_t)$ to perform a weighted sum of $\mathcal{Z}$ to obtain the latent policy feature z_t:

$$z_t = \sum_{k=1}^{K} \mathcal{Z}_k \cdot p_{\theta,k}(z|u_t). \tag{1}$$

This improvement allows us to involve multiple policies within a single response and expand the number of fine-grained policies through combinations.

For the generator G, the policy z_t is first transferred into policy tokens using P-Former. Then policy tokens, along with the dialogue history h_t, are fed into the LLM to guide the generation of the response u_t. By computing the reconstruction loss $\mathcal{L}_{con}$ of G and propagating gradient backward, we can simultaneously optimize E, G, and $\mathcal{Z}$.

After the 1st training, we employ E to annotate pseudo labels $\hat{z}_t$ for each system utterance in $\mathcal{D}$, expanding $\mathcal{D}$ to $\{(h_t, u_t^{sys}, u_t^{usr}, \hat{z}_t, r_t)\}$. Using $\mathcal{D}$, we are able to apply RL algorithm to optimize the policy planning capabilities.

2nd Stage: *Latent Policy Distillation.* To expedite the RL training process in the 3rd stage, we initialize the policy planner P by distilling the utterance encoder E. Specifically, for a response u_t, we compute the predicted policy distributions using E and P as $p_\theta(z_t|u_t)$ and $p_\phi(z_t|h_t)$, respectively. Then we freeze E and minimize the KL divergence to drive P to learn from E. However, we observe that the training set contains many inappropriate system utterances that lead to unsuccessful dialogues, which may harm P's planning ability. Therefore, we use the high-level rewards of each system response for data filtering, denoted as:

$$\begin{aligned}\mathcal{L}_{kl}(\phi) = \sum_{(u_t,h_t,r_t)\in\mathcal{D}} \mathbb{I}(r_t > \delta)\cdot \\ \text{KL\_div}(p_\theta(z|u_t)||p_\phi(z|h_t)),\end{aligned} \tag{2}$$

where $\mathbb{I}(.)$ refers to indicator function. θ and ϕ represent the trainable parameters of E and P. And δ is a predefined threshold. We term this process policy distillation.

To stabilize RL learning, We also initialize the action-value function network Q_α and the value function network V_β at this stage. These two networks evaluate dialogue states during Stage 3, which are implemented by stacking two MLP layers on the policy planner P. To pretrain Q_α and V_β, we use the off-the-shelf Offline RL algorithm IQL (Kostrikov, Nair, and Levine 2021), with the following optimization objectives, respectively:

$$\begin{aligned}\mathcal{L}_V(\beta) &= \mathbb{E}_{(h_t,z_t)\in\mathcal{D}}[L_2^\tau(Q_\alpha(h_t,z_t), V_\beta(h_t))], \\ \mathcal{L}_Q(\alpha) &= \mathbb{E}_{(h_t,z_t,H_{t+1})\in\mathcal{D}}[(r_t + \gamma V_\beta(h_{t+1}) \\ &\qquad - Q_\alpha(h_t,z_t))^2],\end{aligned} \tag{3}$$

where α and β are trainable parameters of Q_α and V_β, respectively. And $\mathcal{L}_2^\tau$ means Expectile Regression Loss (Kostrikov, Nair, and Levine 2021). Therefore, the final optimization objective for this stage is as:

$$\mathcal{L}_2 = \mathcal{L}_{kl}(\phi) + \mathcal{L}_Q(\alpha) + \mathcal{L}_V(\beta). \tag{4}$$

3rd Stage: *Offline Hierarchical RL Enhancement.* To optimize this system only using training data without interactions with simulated environments, we propose an offline hierarchical RL algorithm to learn the policy-level and token-level MDPs simultaneously.

For the policy-level MDP, we utilize the IQL (Kostrikov, Nair, and Levine 2021) to simultaneously train the policy planner P, and the Q-, V-networks. The optimization objectives for the latter two are given by Eq.(3), and the optimization target for the policy planner P is:

$$\begin{aligned}\mathcal{L}_{high}(\phi) = -\mathbb{E}_{(u_t,h_t,z_t)\sim\mathcal{D}}[\exp(\tau(Q_\alpha(h_t,z_t) \\ - V_\beta(h_t)))\log p_\phi(h_t|z_t)],\end{aligned} \tag{5}$$

where $\tau \geq 0$ is the hyperparamter. The motivation behind the optimization target is to apply the advantage function $A(h_t, z_t) = Q_\alpha(h_t, z_t) - V_\beta(h_t)$ to weight each training sample $(u_t, h_t, z_t) \in \mathcal{D}$, thereby enabling selective learning from training data.

For the token-level MDP, we use the REINFORCE algorithm (Sutton et al. 1999) to optimize Generator G, aiming to improve the generation quality. Each intermediate token receives zero reward, and a final reward of $\exp(A(h_t, z_t))$ is given after generating the complete u_t. We optimize G using the following objective:

$$\begin{aligned}\mathcal{L}_{low}(\psi) = -\sum_{(u_t,h_t,z_t)\sim\mathcal{D}} \exp\left(A(h_t,z_t)\right) \\ \cdot \sum_{w_i\in u_t} \log p_\psi(w_i|h_t,z_t,w_{1:i-1}),\end{aligned} \tag{6}$$

where ψ denotes the trainable parameters of Generator G. For proof of this conclusion and empirical explanation, please refer to the appendix. It is important to note that we freeze the parameters of the LLM, so training Generator G actually optimizes the P-Former. Ultimately, the training target of this stage is:

$$\mathcal{L}_3 = \mathcal{L}_{high} + \mathcal{L}_{low} + \mathcal{L}_V + \mathcal{L}_Q. \tag{7}$$

By jointly training the policy planner P and generator G, we simultaneously enhance the system's policy planning capability and the response quality given latent policies.

Policy Usage	Models	ExTES			Generalization to ESConv			P4G		
		SSR↑	SR↑	AvgT↓	SSR↑	SR↑	AvgT↓	SSR↑	SR↑	AvgT↓
Predefined Policy	Proactive	0.544	0.605	7.638	0.430	0.408	7.754	0.012	0.045	7.930
	ProCoT	0.486	0.490	8.128	0.410	0.438	7.992	0.542	0.400	6.885
	PPDPP	0.511	0.558	8.163	0.488	0.515	7.865	0.635	0.745	5.555
No Need for Policy	Standard Prompt									
	+ ChatGPT	0.650	0.810	6.138	0.639	0.762	6.546	0.477	0.460	7.025
	+ Qwen1.5-1.8b	0.538	0.613	7.590	0.543	0.623	6.723	0.683	0.630	6.320
	ICL-AIF	0.474	0.555	7.655	0.542	0.669	6.415	0.063	0.070	7.640
	LoRA Finetuning (32, 64)	0.558	0.627	7.308	0.616	0.662	6.738	0.651	0.655	6.645
	LoRA Finetuning (64, 128)	0.566	0.628	7.450	0.583	0.654	6.892	0.541	0.570	6.830
Automatically Discover Latent Policy	LDPP	**0.723**	**0.903**	**4.132**	**0.651**	**0.781**	**5.388**	**0.733**	**0.795**	**5.570**
	-w/o *2nd Stage*	0.716	0.865	4.483	0.637	0.769	5.608	0.715	0.760	6.140
	-w/o *3rd Stage*	0.560	0.623	7.038	0.528	0.538	7.777	0.550	0.570	6.840

Table 1: Main results on ExTES, ESConv, and P4G, using gpt-3.5-turbo-0613 as the critic. LoRA Fine-tuning(x, y) means setting *lora rank*=x and *lora alpha*=y. Results on ESConv are conducted using the planner trained on ExTES.

Experimental Settings

Datasets

We evaluate the proposed framework on two typical applications of proactive dialogues, ExTES (Zheng et al. 2023b) (emotional support) and P4G (Wang et al. 2019a) (persuasion), representing collaborative and non-collaborative dialogue, respectively. ExTES is an extension of ESConv (Liu et al. 2021b), comprising sufficient dialogues for training (11,117 complete dialogues). We randomly divide it into 10,717/200/200 for train/valid/test set. P4G includes 1,017 donation persuasion dialogues where a "persuader" attempts to persuade a "persuadee" to donate to a charity called Save the Children. We randomly choose 100/100 dialogues for validation/testing. We take the remaining 817 dialogues as the training set. In practice, we extend the training set of dialogues to 5,579 using ChatGPT (Ouyang et al. 2022) due to the limited size. Please see the appendix for details of data augmentation. Given that ExTES is larger than P4G and P4G contains synthetic data, ExTES is more suitable for our task setup. Consequently, our primary analysis and experiments were conducted on ExTES. **Furthermore, to evaluate the generalizability of LDPP, we also test on ESConv (130 test cases) using LDPP trained on ExTES.**

Baselines

We compare Proactive (Deng et al. 2023a), ProCoT (Deng et al. 2023a), and PPDPP (Deng et al. 2023b) for baselines in need of predefined policies. Proactive and ProCoT require LLMs to select the most appropriate strategy before replying. PPDPP learns a specialized policy planner based on the predefined policies. For methods not requiring policy use, we select the standard prompt method (prompting the base LLM to generate replies directly without considering dialogue policies), LoRA-based fine-tuning (Hu et al. 2021) (shorted as LoRA), and ICL-AIF (Fu et al. 2023). ICL-AIF prompts LLMs to provide suggestions before generating corresponding responses.

Evaluation Methods

Self-play evaluation. Since correct policies are often not unique and the absence of explicitly defined policies in our settings, directly assessing policy prediction accuracy is infeasible. We follow the same self-play method as previous work (Deng et al. 2023b) for dialogue-level evaluation. Specifically, two LLMs simulate the system and user in multi-turn dialogues, with the system receiving strategy guidance from a planner. We also prompt an LLM as critic to evaluate the completion status of each turn, deeming the dialogue failed if the goal isn't met within 10 turns. For more detailed prompts, please refer to the appendix.

Critic model. We also use ChatGPT to assess dialogue completion status following PPDPP. For ExTES and ESConv, we define four states: [worse, same, better, solved], with corresponding rewards of [-1, -0.5, 0.1, 1.0]; for P4G, the states are [reject, neutral, positive, donate], with also rewards of [-1, -0.5, 0.1, 1.0]. ChatGPT classifies the current dialogue state into one of 4 states. We perform 10 times classification per evaluation to reduce randomness, with each time getting a scalar value. We average them to obtain the reward r_t for the current dialogue turn. A dialogue is considered successful if $r_t > \eta$. We set $\eta = 0.6$ instead of 0.1 in PPDPP to improve the robustness of evaluations. **To ensure the robustness of results, we run the main experiments at least twice and reported the average results. To further reduce evaluation bias, we use two versions of ChatGPT (gpt-3.5-turbo-0613 and -0125) to serve as critics and present the latter results in the appendix.**

Metrics. Following Deng et al. (2023b), we use two common dialogue-level metrics: Success Rate (SR) and Average Turn (AvgT). SR measures effectiveness and is defined as the ratio of the number of successful cases to the total number of test cases. AvgT measures the efficiency of goal completion by calculating the average dialogue turns of all test cases. However, we observe the high variance of SR. Therefore, we introduce the **SSR metric to more accurately assess effectiveness**. SSR complements the SR, where SR calculates the ratio of success by mapping the final turn reward

(a) ExTES

LDPP	Ide.		Com.		Sug.		Ove.	
vs.	Win	Lose	Win	Lose	Win	Lose	Win	Lose
PPDPP	8%	8%	52%	6%	64%	8%	68%	10%
LoRA	6%	32%	6%	6%	32%	10%	26%	18%

(b) P4G

LDPP	Inf.		Per.		Ove.	
vs.	Win	Lose	Win	Lose	Win	Lose
PPDPP	32%	20%	40%	26%	48%	22%
LoRA	10%	14%	24%	16%	26%	16%

Table 2: Human evaluation results on ExTES and P4G.

into a binary 0 or 1 while SSR averages all final turn rewards directly. Therefore, we view SSR as a "Soft SR".

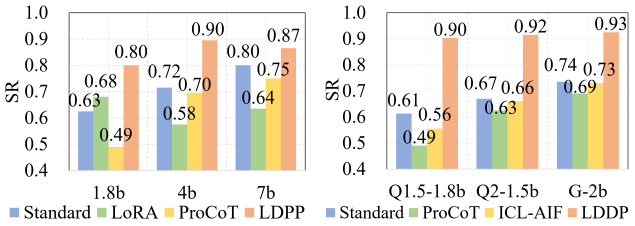
Backbone. We conduct main experiments based on Qwen1.5-1.8b (Bai et al. 2023) and analysis studies on a series of LLMs: Qwen1.5-1.8b, -4b, -7b, Qwen2-1.5b, and Gemma-2b (Mesnard et al. 2024). Due to the hardware limitations, we select models under 7B parameters. We employ these LLMs to play the roles of Therapist/Persuader, respectively, guided by policies from the planner.

Results and Analysis

Main Results

Based on Table 1, we find that *LDPP outperforms all baselines significantly on all tasks*. This LDPP is implemented with $(T, L, K) = (8, 6, 24)$. Firstly, LDPP achieves notable enhancements compared to the standard prompt and LoRA methods, verifying the effectiveness of latent policies and the P-Former module. Prompt-based methods like Proactive, ProCoT, and ICL-AIF show unsatisfactory and unstable performance. We observe serious role confusion issues in these works. Due to the disturbance of suggestions or analyses, the system's responses fail to meet the expected form, leading to the role confusion during dialogue. We attribute this to the limited instruction-following and analysis capabilities of the 1.8b LLM. Compared to PPDPP, LDPP performs more effectively and more efficiently without online learning and predefined policies, proving the effectiveness of self-supervised policy discovery and offline hierarchical RL training method. Besides, we also find that *LDPP based on Qwen1.5-1.8B performs better than ChatGPT*, further affirming our method's effectiveness. This also demonstrates that, with the assistance of external modules, smaller LLMs can surpass larger ones. For more results with the different LLM as critic, please refer to Table 6 in the appendix.

Furthermore, we conduct ablation experiments by skipping Stage 2 and Stage 3. Firstly, the significant performance drop without Stage 3 underscores its necessity for learning policy planning. Without Stage 3, the policy planner can only learn from the utterance encoder, thus failing to acquire planning capabilities. Besides, the slight decline observed without Stage 2 also shows the rationality for proper initialization for effective RL-based policy planning.



(a) LLM Size (b) LLM Series

Figure 2: Performance comparison as the LLM size and LLM series change on ExTES. Q1.5-1.8b, Q2-1.5b, and G-2b refer to Qwen1.5-1.8b, Qwen2-1.5b, and Gemma-2b.

Human Evaluation

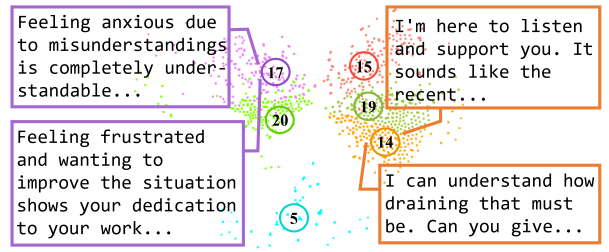
Following previous studies (He et al. 2024), we conduct human evaluation on 50 dialogues randomly sampled from the test in ExTES and P4G, respectively. We selected two training-based baselines, PPDPP and LoRA, based on whether they require predefined policies and a simulated environment. Three annotators are required to compare the dialogues generated by LDPP/PPDPP and LDPP/LoRA. We assess four metrics: **Identification (Ide.)**, **Comforting (Com.)**, **Suggestion (Sug.)**, and **Overall (Ove.)** for ExTES and three metrics: **Information (Inf,)**, **Persuasion (Per.)**, and **Overall (Ove.)** for P4G. Detailed instructions for the annotators are provided in the appendix. Results are presented in Table 2. First, LDPP outperforms PPDPP and LoRA in the **Ove.** metric, aligning with results in Table 1. We observe that LDPP does not like to ask patients for specific details, often providing suggestions quickly after the patient's introduction. While providing useful suggestions is crucial and could improve SR evaluation, failing to conduct thorough inquiries impacts the practical experience. To alleviate this phenomenon, designing relevant rewards could be helpful.

Performance on Different LLMs

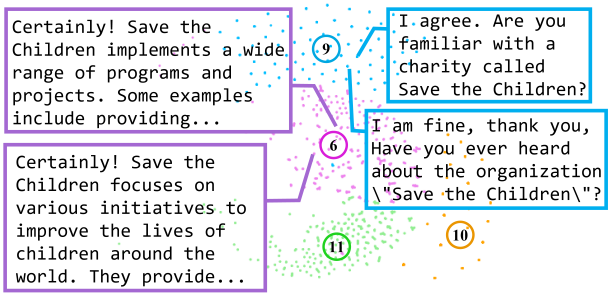
To further validate our proposed framework, we conduct experiments on LLMs with different sizes. Specifically, we compare LDPPs based on Qwen1.5-1.8b, 4b, and 7b for different sizes with settings of (T, L, K)=$(8, 4, 24)$. The results are shown in Figure 2. We observe that *LDPP achieves the best performance in all three different sizes*. As LLM size increases, standard prompting and prompting-based method ProCoT show continuous improvement, but they still perform worse than LDPP. In contrast, LoRA Fine-tuning exhibits significant variability. The reason may be that fine-tuning fails to differentiate data quality and train the added parameters sufficiently, harming LLMs' generalization ability. Besides, we also conduct experiments ((T, L, K)=$(8, 6, 24)$) using Qwen1.5-1.8b, Qwen2-1.5b, and Gemma-2b for different LLM series and present in Figure 2, we find that LDPP also performs best.

Latent Policy Visualization

To intuitively demonstrate the learned latent policies, we visualize the policies of system utterances in Figure 3. Initially, each utterance is encoded into a latent policy feature



(a) ExTES



(b) P4G

Figure 3: Visualization of latent policies for utterances belong to top-4 and top-6 most frequently used policies.

	K=6	K=12	K=18	K=24
SSR	0.687	0.652	0.675	0.628
AvgT	4.50	5.84	4.86	5.77

Table 3: Results of different K on ExTES.

following Eq.(1) and classified into the closest policy vector in the Codebook. For each policy vector in the Codebook, we select the top-500 closest latent policy features and then apply PCA for dimensionality reduction on them to draw a scatter plot. For ease of presentation, we only display those from the 6/4 most frequently used policy vectors for ExTES/P4G. We also present parts of text utterances for comparison and observe that utterances within the same cluster are indeed semantically similar, validating the effectiveness of stage 1. To better understand these policies, Table 12 in the appendix presents three representative utterances for each of them. These utterance examples can help to understand the semantical operations for policies in the Codebook.

Parameter Sensitivity Analysis

Codebook Size K. We investigate the impact of Codebook size K on guiding the proactive dialogue process. Experiments are conducted on the ExTES dataset with K= 6, 12, 18, and 24, while keeping other hyper-parameters constant $(T = 8, L = 4)$, as shown in Table 3. LDPP achieves relatively stable results and performs satisfactorily even with the smallest K, which can be attributed to the method of capturing latent policy features: by computing a weighted sum of the Codebook based on the policy distribution derived from the policy planner, it allows for a semantic combination of different policy vectors within the Codebook. Therefore, even with a small K, a wide range of latent policies can be expressed. However, performance decreased when $K = 24$. We assume that this is due to the increased complexity of predicting the distribution for the larger Codebook, which requires additional training steps.

	T=2	T=8	T=16	T=24
SSR	0.699	0.628	0.619	0.628
AvgT	4.57	5.77	6.17	5.65

Table 4: Results of different #policy tokens (T) on ExTES.

	ExTES			**P4G**		
	L=2	L=4	L=6	L=2	L=4	L=6
SSR	0.649	0.628	0.719	0.580	0.711	0.732
AvgT	5.42	5.77	3.88	6.37	5.85	5.49

Table 5: Results of different P-Former layers (L).

#Policy Tokens T. Although we aim for these policy tokens to align with the input word embeddings of LLMs, they do not inherently belong to the LLMs' vocabulary. Therefore, it is important to analyze the potential noise introduced by the policy tokens into the LLMs. We set T as 2, 8, 16, and 24 while keeping $(L = 4, K = 24)$. The experimental results are presented in Table 4. Overall, there is a trend of decreasing dialogue success rate as T increases, indicating that a greater number of policy tokens indeed introduce noise, adversely affecting response generation. In practice, users can reduce the number of query tokens or enhance the capacity of P-Former (e.g., increasing the number of layers) to mitigate the impact of noise.

P-Former Layer L. The number of P-Former layers reflects its parameter size and capability. We hypothesize that a stronger P-Former reduces the gap between the transferred policy tokens and the LLMs' input space while retaining more policy semantic information. To validate this, we set different layers on ExTES and P4G. The results, presented in Table 5, indicate that the number of P-Former layers impacts dialogue performances, especially on P4G, where more layers notably improve the dialogue success rate. On the P4G dataset, we observed zero improvement. This indicates that only if the P-Former is sufficiently powerful can we effectively utilize the latent policy.

Conclusion and Future Work

In this work, we introduce a novel learning scenario that discovers potential policies from broadly collected dialogue records and learns policy planning without dynamic interactions with simulated environments. To address this challenge, we propose a new learning framework called LDPP, containing three stages: latent policy discovery, policy distillation, and offline RL enhancement. Experimental results demonstrate that LDPP significantly improves LLMs' proactive dialogue capabilities, achieving more pronounced and consistent enhancements compared to all baselines, even ChatGPT. Future research will mainly focus on improving the explainability of latent policies, ensuring the reliability of policies used in proactive dialogue.

Acknowledgements

The research in this article is supported by the National Science Foundation of China (U22B2059, 62276083), the Human-Machine Integrated Consultation System for Cardiovascular Diseases (2023A003). We also appreciate the support from China Mobile Group Heilongjiang Co., Ltd. @ on our research,the research is jointly completed by both parties. This research was also supported by the Google South Asia & Southeast Asia research award. We are sincerely grateful to all reviewers for their insightful feedback.

References

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; Hui, B.; Ji, L.; Li, M.; Lin, J.; Lin, R.; Liu, D.; Liu, G.; Lu, C.; Lu, K.; Ma, J.; Men, R.; Ren, X.; Ren, X.; Tan, C.; Tan, S.; Tu, J.; Wang, P.; Wang, S.; Wang, W.; Wu, S.; Xu, B.; Xu, J.; Yang, A.; Yang, H.; Yang, J.; Yang, S.; Yao, Y.; Yu, B.; Yuan, H.; Yuan, Z.; Zhang, J.; Zhang, X.; Zhang, Y.; Zhang, Z.; Zhou, C.; Zhou, J.; Zhou, X.; and Zhu, T. 2023. Qwen Technical Report. *arXiv preprint arXiv:2309.16609*.

Bao, S.; He, H.; Wang, F.; and Wu, H. 2019. PLATO: Pre-trained Dialogue Generation Model with Discrete Latent Variable. In *Annual Meeting of the Association for Computational Linguistics*.

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Józefowicz, R.; and Bengio, S. 2015. Generating Sentences from a Continuous Space. In *Conference on Computational Natural Language Learning*.

Chen, M.; Yu, X.; Shi, W.; Awasthi, U.; and Yu, Z. 2023. Controllable mixed-initiative dialogue generation through prompting. *arXiv preprint arXiv:2305.04147*.

Chen, W.; Gong, Y.; Wang, S.; Yao, B.; Qi, W.; Wei, Z.; Hu, X.-M.; Zhou, B.; Mao, Y.; Chen, W.; Cheng, B.; and Duan, N. 2022. DialogVED: A Pre-trained Latent Variable Encoder-Decoder Model for Dialog Response Generation. In *Annual Meeting of the Association for Computational Linguistics*.

Cheng, Y.; Liu, W.; Li, W.; Wang, J.; Zhao, R.; Liu, B.; Liang, X.; and Zheng, Y. 2022. Improving multi-turn emotional support dialogue generation with lookahead strategy planning. *arXiv preprint arXiv:2210.04242*.

Cheng, Y.; Liu, W.; Wang, J.; Leong, C. T.; Ouyang, Y.; Li, W.; Wu, X.; and Zheng, Y. 2024. Cooper: Coordinating specialized agents towards a complex dialogue goal. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17853–17861.

Cho, I.; Takahashi, R.; Yanase, Y.; and Saito, H. 2023. Deep RL with Hierarchical Action Exploration for Dialogue Generation. *ArXiv*, abs/2303.13465.

Deng, Y.; Lei, W.; Liao, L.; and Chua, T.-S. 2023a. Prompting and Evaluating Large Language Models for Proactive Dialogues: Clarification, Target-guided, and Non-collaboration. *arXiv preprint arXiv:2305.13626*.

Deng, Y.; Zhang, W.; Lam, W.; Ng, S.-K.; and Chua, T.-S. 2023b. Plug-and-play policy planner for large language model powered dialogue agents. In *The Twelfth International Conference on Learning Representations*.

Fu, Y.; Peng, H.; Khot, T.; and Lapata, M. 2023. Improving language model negotiation with self-play and in-context learning from ai feedback. *arXiv preprint arXiv:2305.10142*.

He, H.; Chen, D.; Balakrishnan, A.; and Liang, P. 2018. Decoupling Strategy and Generation in Negotiation Dialogues. *ArXiv*, abs/1808.09637.

He, T.; Liao, L.; Cao, Y.; Liu, Y.; Liu, M.; Chen, Z.; and Qin, B. 2024. Planning Like Human: A Dual-process Framework for Dialogue Planning. *arXiv preprint arXiv:2406.05374*.

Hu, J. E.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *ArXiv*, abs/2106.09685.

Hu, Z.; Feng, Y.; Deng, Y.; Li, Z.; Ng, S.-K.; Luu, A. T.; and Hooi, B. 2023. Enhancing Large Language Model Induced Task-Oriented Dialogue Systems Through Look-Forward Motivated Goals. *arXiv preprint arXiv:2309.08949*.

Kahneman, D. 2003. Maps of Bounded Rationality: Psychology for Behavioral Economics. *The American Economic Review*, 93: 1449–1475.

Kang, D.; Kim, S.; Kwon, T.; Moon, S.; Cho, H.; Yu, Y.; Lee, D.; and Yeo, J. 2024. Can Large Language Models be Good Emotional Supporter? Mitigating Preference Bias on Emotional Support Conversation. *arXiv preprint arXiv:2402.13211*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kostrikov, I.; Nair, A.; and Levine, S. 2021. Offline Reinforcement Learning with Implicit Q-Learning. *ArXiv*, abs/2110.06169.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. C. H. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *International Conference on Machine Learning*.

Liebana, D. P.; Dieskau, J.; Hunermund, M.; Mostaghim, S.; and Lucas, S. M. M. 2015. Open Loop Search for General Video Game Playing. *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation.*

Liu, A.; Wang, B.; Tan, Y.; Zhao, D.; Huang, K.; He, R.; and Hou, Y. 2023. MTGP: Multi-turn Target-oriented Dialogue Guided by Generative Global Path with Flexible Turns. In *Findings of the Association for Computational Linguistics: ACL 2023*, 259–271.

Liu, J.; Pan, F.; and Luo, L. 2020. GoChat: Goal-oriented Chatbots with Hierarchical Reinforcement Learning. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Liu, S.; Zheng, C.; Demasi, O.; Sabour, S.; Li, Y.; Yu, Z.; Jiang, Y.; and Huang, M. 2021a. Towards emotional support dialog systems. *arXiv preprint arXiv:2106.01144*.

Liu, S.; Zheng, C.; Demasi, O.; Sabour, S.; Li, Y.; Yu, Z.; Jiang, Y.; and Huang, M. 2021b. Towards Emotional Support Dialog Systems. In *Annual Meeting of the Association for Computational Linguistics*.

Liu, W.; Cheng, Y.; Wang, H.; Tang, J.; Liu, Y.; Zhao, R.; Li, W.; Zheng, Y.; and Liang, X. 2022. " My nose is running."" Are you also coughing?": Building A Medical Diagnosis Agent with Interpretable Inquiry Logics. *arXiv preprint arXiv:2204.13953*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, abs/1907.11692.

Lubis, N.; Geishauser, C.; Heck, M.; Lin, H.-C.; Moresi, M.; van Niekerk, C.; and Gavsi'c, M. 2020. LAVA: Latent Action Spaces via Variational Auto-encoding for Dialogue Policy Optimization. *ArXiv*, abs/2011.09378.

Mesnard, G. T. T.; Hardin, C.; Dadashi, R.; Bhupatiraju, S.; Pathak, S.; Sifre, L.; Riviere, M.; Kale, M.; Love, J. C.; Tafti, P. D.; Hussenot, L.; Chowdhery, A.; Roberts, A.; Barua, A.; Botev, A.; Castro-Ros, A.; Slone, A.; H'eliou, A.; Tacchetti, A.; Bulanova, A.; Paterson, A.; Tsai, B.; Shahriari, B.; Lan, C. L.; Choquette-Choo, C. A.; Crepy, C.; Cer, D.; Ippolito, D.; Reid, D.; Buchatskaya, E.; Ni, E.; Noland, E.; Yan, G.; Tucker, G.; Muraru, G.-C.; Rozhdestvenskiy, G.; Michalewski, H.; Tenney, I.; Grishchenko, I.; Austin, J.; Keeling, J.; Labanowski, J.; Lespiau, J.-B.; Stanway, J.; Brennan, J.; Chen, J.; Ferret, J.; Chiu, J.; Mao-Jones, J.; Lee, K.; Yu, K.; Millican, K.; Sjoesund, L. L.; Lee, L.; Dixon, L.; Reid, M.; Mikula, M.; Wirth, M.; Sharman, M.; Chinaev, N.; Thain, N.; Bachem, O.; Chang, O.; Wahltinez, O.; Bailey, P.; Michel, P.; Yotov, P.; Sessa, P. G.; Chaabouni, R.; Comanescu, R.; Jana, R.; Anil, R.; McIlroy, R.; Liu, R.; Mullins, R.; Smith, S. L.; Borgeaud, S.; Girgin, S.; Douglas, S.; Pandya, S.; Shakeri, S.; De, S.; Klimenko, T.; Hennigan, T.; Feinberg, V.; Stokowiec, W.; hui Chen, Y.; Ahmed, Z.; Gong, Z.; Warkentin, T. B.; Peran, L.; Giang, M.; Farabet, C.; Vinyals, O.; Dean, J.; Kavukcuoglu, K.; Hassabis, D.; Ghahramani, Z.; Eck, D.; Barral, J.; Pereira, F.; Collins, E.; Joulin, A.; Fiedel, N.; Senter, E.; Andreev, A.; and Kenealy, K. 2024. Gemma: Open Models Based on Gemini Research and Technology. *ArXiv*, abs/2403.08295.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L. E.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. J. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Samad, A. M.; Mishra, K.; Firdaus, M.; and Ekbal, A. 2022. Empathetic persuasion: reinforcing empathy and persuasiveness in dialogue systems. In *Findings of the Association for Computational Linguistics: NAACL 2022*, 844–856.

Song, I.; Pendse, S. R.; Kumar, N.; and De Choudhury, M. 2024. The typing cure: Experiences with large language model chatbots for mental health support. *arXiv preprint arXiv:2401.14362*.

Sutton, R. S.; McAllester, D. A.; Singh, S.; and Mansour, Y. 1999. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Neural Information Processing Systems*.

van den Oord, A.; Kalchbrenner, N.; and Kavukcuoglu, K. 2016. Pixel Recurrent Neural Networks. In *International Conference on Machine Learning*.

Wang, J.; Zhang, Y.; Kim, T.-K.; and Gu, Y. 2020. Modelling Hierarchical Structure between Dialogue Policy and Natural Language Generator with Option Framework for Task-oriented Dialogue System. *ArXiv*, abs/2006.06814.

Wang, X.; Shi, W.; Kim, R.; Oh, Y.; Yang, S.; Zhang, J.; and Yu, Z. 2019a. Persuasion for good: Towards a personalized persuasive dialogue system for social good. *arXiv preprint arXiv:1906.06725*.

Wang, X.; Shi, W.; Kim, R.; Oh, Y. J.; Yang, S.; Zhang, J.; and Yu, Z. 2019b. Persuasion for Good: Towards a Personalized Persuasive Dialogue System for Social Good. *ArXiv*, abs/1906.06725.

Xiang, J.; Liu, Z.; Liu, H.; Bai, Y.; Cheng, J.; and Chen, W. 2024. DiffusionDialog: A Diffusion Model for Diverse Dialog Generation with Latent Space. In *International Conference on Language Resources and Evaluation*.

Yang, Y.; Li, Y.; and Quan, X. 2021. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 14230–14238.

Yu, X.; Chen, M.; and Yu, Z. 2023. Prompt-Based Monte-Carlo Tree Search for Goal-Oriented Dialogue Policy Planning. In *Conference on Empirical Methods in Natural Language Processing*.

Zhang, Q.; Naradowsky, J.; and Miyao, Y. 2023. Ask an Expert: Leveraging Language Models to Improve Strategic Reasoning in Goal-Oriented Dialogue Models. In *Annual Meeting of the Association for Computational Linguistics*.

Zhang, T.; Huang, C.; Deng, Y.; Liang, H.; Liu, J.; Wen, Z.; Lei, W.; and Chua, T.-S. 2024. Strength Lies in Differences! Towards Effective Non-collaborative Dialogues via Tailored Strategy Planning. *arXiv preprint arXiv:2403.06769*.

Zhao, T.; Zhao, R.; and Eskénazi, M. 2017. Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders. In *Annual Meeting of the Association for Computational Linguistics*.

Zhao, W.; Zhao, Y.; Lu, X.; Wang, S.; Tong, Y.; and Qin, B. 2023. Is ChatGPT equipped with emotional dialogue capabilities? *arXiv preprint arXiv:2304.09582*.

Zheng, Z.; Liao, L.; Deng, Y.; and Nie, L. 2023a. Building Emotional Support Chatbots in the Era of LLMs. *ArXiv*, abs/2308.11584.

Zheng, Z.; Liao, L.; Deng, Y.; and Nie, L. 2023b. Building emotional support chatbots in the era of llms. *arXiv preprint arXiv:2308.11584*.

Zhou, Y.; He, H.; Black, A. W.; and Tsvetkov, Y. 2019. A dynamic strategy coach for effective negotiation. *arXiv preprint arXiv:1909.13426*.

Zhou, Y.; Zanette, A.; Pan, J.; Levine, S.; and Kumar, A. 2024. ArCHer: Training Language Model Agents via Hierarchical Multi-Turn RL. *arXiv preprint arXiv:2402.19446*.