# LLM-DDP: Improving Policy Learning for Composite Task Oriented Dialogues via Large Language Model feedbacks

**Anonymous ACL submission**

## Abstract

Dialogue Policy (DP) is pivotal in Task-Oriented Dialogue (TOD), and Reinforcement Learning (RL) has shown good effectiveness in training DP scenarios. However, RL-based DPs encounter challenges in composite tasks with domain dependencies, which involve managing interrelated subtasks across various domains. In particular, the Proximal Policy Optimization (PPO) algorithm, as an efficient, stable, and user-friendly RL algorithm, is gradually becoming the preferred tool for solving complex reinforcement learning tasks; meanwhile, Large Language Models (LLMs) have shown a profound understanding of common sense content across various domains. Therefore, we propose the integration of LLMs with an enhanced PPO method to tackle composite tasks, which we term the LLM Feedback Domain Dependent Policy (LLM-DDP). Improving the capability of TOD systems to address domain-dependent issues is achieved by integrating the domain prioritization logic of LLMs into the actor-critic framework of PPO. Furthermore, we introduce a domain-driven critic loss function, which enhances the policy network's ability to incorporate domain prioritization logic. In the MultiWOZ 2.1 dataset, with identical parameter configurations and dialogue turns, our study achieved superior performance and validated the efficacy of the proposed methodology.

## 1 Introduction

Task-Oriented Dialogue (TOD) is a type of dialogue system designed to facilitate the completion of specific tasks or actions. It emphasizes the efficiency and accuracy of user interactions, with the aim of achieving the desired outcomes with the fewest number of sessions. Over the years of development, numerous outstanding systems have emerged in the field of task-oriented multi-turn dialogue, such as LaMDA, Senseforth, and Cognigy.

TOD systems typically adopt two structural paradigms: end-to-end and pipeline architectures. End-to-end TOD adopts an approach of holistic system modeling. However, due to the model's inherent black-box nature, its decision-making processes and generated responses may be difficult to control, improve, and optimize. In contrast, the pipeline architecture composed of individual modules enables researchers to take advantage of diverse technical means to improve the efficiency of development and debugging processes. The architectural pipeline of a TOD system is primarily constituted by four core components: Natural Language Understanding (NLU) (Wang et al., 2022; Mirza et al., 2024), Dialogue State Tracking (DST) (Balaraman et al., 2021), Dialogue Policy (DP) and Natural Language Generation (NLG) (Ohashi and Higashinaka, 2022a). The DP module plays a crucial role in TOD. It is responsible for making flexible and reasonable decisions regarding the system's next actions based on the current dialogue state, which includes aspects such as user requirements, provided information, and the progress of the task. This aims to optimize the dialogue process, improve task completion efficiency, and balance user satisfaction.

Dialogue tasks can be categorized into three types according to their complexity and domain scope: single domain tasks, multi-domain tasks, and composite tasks (Balaraman and Magnini, 2021; Peng et al., 2017). Single-domain tasks, such as weather inquiries, relate to a single domain of expertise. Multi-domain tasks, on the other hand, involve the system managing multiple tasks concurrently, such as providing news updates and querying weather information. In these scenarios, each task domain is independent and unrelated. Composite tasks may span multiple domains and require switching between dependent domains (Peng et al., 2017). For example, the task of travel planning encompasses domains such as train, hotel, interesting,

and restaurants. These domains have dependency relationships. Specifically, booking train tickets first may make it impossible to reserve a suitable hotel, or reserving a restaurant first may result in the inability to find an appropriate hotel option. Performing analysis and evaluation of the dependencies between various domains in the DP module can significantly reduce the number of interaction turns between the system and the user, thus substantially enhancing the efficiency and success rate of dialogue.

In the type of composite domain tasks, current approaches to DP are inadequate to capture intricate interdependencies across various domains, and some scholars are working to improve this issue based on Reinforcement Learning (RL). Wang et al. (2020a) proposed modeling the hierarchical structure between DP and NLG with the option framework, where the latent dialogue act is applied to avoid designing specific dialogue act representations. (Zhao et al., 2024) proposed a novel Bootstrapped Policy Learning framework that adaptively tailors curricula for complex goals through goal design with progressively challenging subgoals, combines these aspects to enable smooth knowledge transitions from simple to complex goals, enhancing the learning efficiency of DP. However, the design of task hierarchies requires substantial domain knowledge and is inherently unstable, further complicating the resolution of domain dependencies.

Existing experiments have shown that the PPO algorithm achieves a relatively high success rate among RL algorithms. Meanwhile, large language models (LLMs) possess common sense knowledge and strong semantic comprehension abilities. The combination of PPO and LLMs opens up new possibilities for addressing the inter-domain dependency issues in complex domain tasks. We integrate these two methodologies and propose a novel method, termed LLM Feedback Domain Dependent Policy (LLM-DDP), which leverages the strengths of both PPO and LLM to enhance decision-making in domain-dependent policies. Initially, to train the DP, we employ Imitation Learning, swiftly attaining proficiency in emulating expert behavior. Subsequently, we introduce large language models to judge domain priorities according to the current dialogue state and filter out unnecessary candidate system actions to reduce the action space. Furthermore, this study introduces a domain-driven Critic loss function, aimed at continuously improving the performance of the policy network in tackling composite tasks. Through the above methods, LLM-DDP reduces the number of dialogue turns, increases the success rate of dialogs, and addresses domain-dependent issues.

## 2 Related Work

### 2.1 Multi-Domain Challenges in Dialogue Policy

To tackle the difficulties of inter-domain dependency, many previous studies have turned to transfer learning as a solution. (Pan and Yang, 2009) takes advantage of shared characteristics and relationships between different tasks, allowing the transfer of knowledge learned from one task to another. However, the selection of appropriate domains itself presents a complex issue. Transfer learning often requires the design of sophisticated model architectures and training strategies, which can significantly increase computational and debugging costs. Wu et al. (2019) introduced a Transferable Dialogue State Generator (TRADE), which enables domain adaptation in zero-shot settings by leveraging knowledge learned from other domains to track slot values in new domains. Meanwhile, Kaiser et al. (2017) proposed the Multi-Model architecture, which converts inputs from different domains into a unified representation through specialized modality nets, allowing the model to handle tasks across multiple domains simultaneously.

In the recent past, some studies have adopted the Hierarchical Reinforcement Learning (HRL) method to address these problems. (Zhu et al., 2023; Rohmatillah and Chien, 2023) decomposes a complex task horizontally or vertically into multiple subtasks that are then executed by different agents. However, this hierarchical structure is inherently unstable. As lower-level policies continuously evolve, transition functions at the higher level also undergo constant changes. Consequently, HRL struggles to address the issue of domain dependency in complex multi-domain tasks.

Unlike the aforementioned studies, this study uses the prior knowledge and semantic comprehension capabilities of LLMs to evaluate domain priorities and quantify the loss function to address domain dependency issues.

### 2.2 LLMs in TOD Systems

The application of LLMs in TOD systems is primarily categorized into two approaches: end-to-

end and pipeline-based. Although end-to-end models offer a more straightforward approach to generating responses (Lee, 2021; Yang et al., 2021), their black-box-like processing may engender limitations in terms of flexibility and maintainability.

Regarding the pipeline-based TOD system, within the NLU module, Yoshimaru et al. (2023) propose a framework that uses LLM asynchronously in the part of the system that returns an appropriate response and in the part that understands the intention of the user to search the database. In the DST module, Gao et al. (2023) employed SOLOIST, a model initialized with pre-trained weights, and subsequently fine-tuned it on a small amount of data obtained from the section. After fine-tuning, the model generates only the domain and slot of the belief state. In the DP module, Kwan et al. (2024) uses a text-to-text Transformer-based model to generate flexible dialogue actions and employs reinforcement learning with a reward-shaping mechanism to efficiently fine-tune the word-level dialogue policy. In the NLG module, Xu et al. (2024) used an LLM to rephrase dialogues, thus generating natural language that is more natural and empathetic.

Although LLMs are currently widely applied to end-to-end TOD systems and in NLU, DST, and NLG modules in pipeline-based TOD systems, research on improving DP using LLMs is relatively scarce. Our approach, which integrates the output of LLMs with the novel loss function, appears to be a relatively novel endeavor in the academic community.

### 2.3 Loss Objective for Dialogue Policy

In the machine learning literature, the cross-entropy (CE) loss function is one of the most widely used optimization objectives so far. However, it faces challenges in application scenarios such as DP training, as it is not robust when dealing with highly imbalanced datasets Lin et al. (2023). Furthermore, mean squared error (MSE) also has limitations when used individually, such as sensitivity to outliers, vanishing or explosion of gradients, and unsuitability for classification problems. As a result, some studies investigate various combinations of loss functions. Wu et al. (2023) trained the model during the DP training phase by combining policy loss and response loss, which led to improved performance. Rohmatillah and Chien (2023) integrated three loss functions from the policy network and the auxiliary network in classification prediction, effectively training and optimizing DP. These approaches have achieved effective results in dialogue tasks of composed domains.

In our study, we integrate two different loss functions to develop a novel composite loss function, aiming to address the issue of efficiently training the DP module in scenarios involving composed domains.

## 3 Method

The LLM-DDP architecture diagram is shown in Figure 1. The approach is implemented based on the ToD pipeline, more details of which can be found in Appendix A.The approach is primarily composed of five key components: (1) Imitation Learning: Pre-train a policy network to accelerate the training process. (2) Action Probability Sampling: Apply the Heaviside Step Function to discretize the predicted probabilities into binary values (0 or 1). (3) Domain Priority Ranking: Obtain prioritization of domains by harnessing the capabilities of LLMs, which is used to produce prioritizations of the next turn's system action. (4) Multi-cross-entropy loss function: Utilize the multi-cross-entropy loss function to guide the policy network toward convergence. (5) Domain-Driven Loss Function for the Critic Module in PPO: Propose a novel loss function formulated for the critic by integrating the stability of MSE and the domain sensitivity of multi-cross-entropy.

### 3.1 Background

In this study, an effective improvement of DP is achieved through the modification of the Actor-Critic framework of the PPO algorithm. It consists of two primary components: the Actor and the Critic. The Actor, a neural network, is tasked with selecting actions according to the current policy. The Critic, another neural network, is responsible for assessing the quality of the Actor's actions by providing a score.

The interaction between the agent (Dialogue System) and the environment (User Context) is formalized using a finite Markov Decision Process (MDP) denoted as ($S$, $A$, $P$, $R$). Here, $S$ represents the set of states, A denotes the set of discrete actions, and $R$ means the set of rewards. At time step $t$, the agent is in a state $S_t \in S$.

### 3.2 Imitation Learning

Initially, we employed Behavior Cloning (BC) to train a policy network that produces actions that

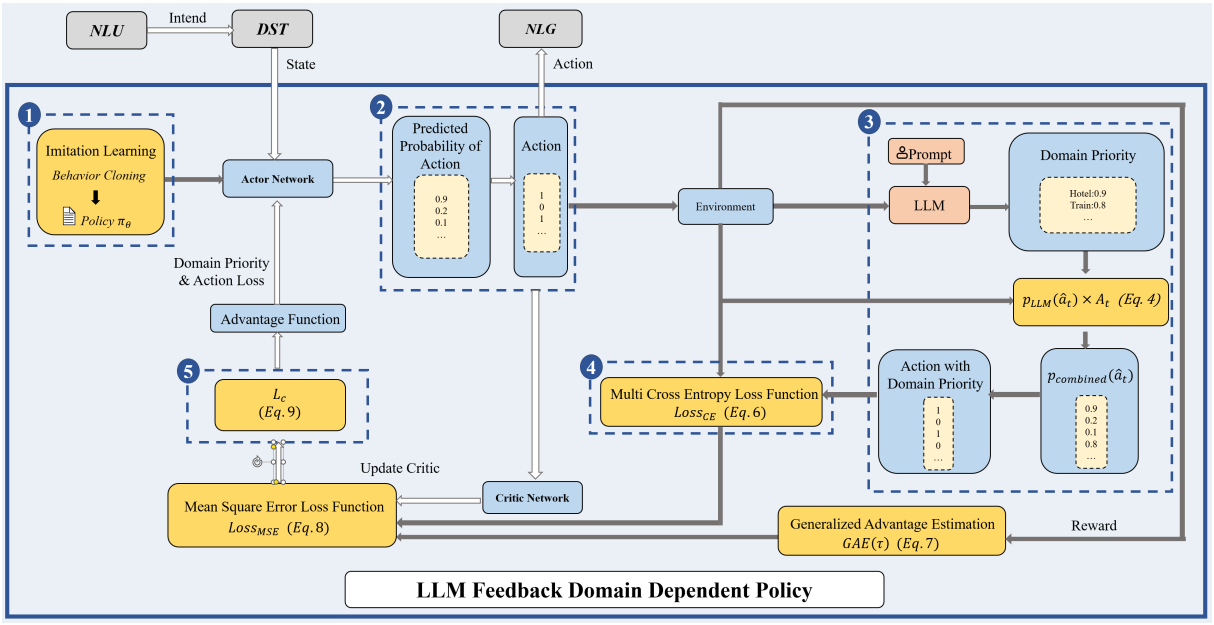


Figure 1: Overview of LLM Feedback Domain Dependent Policy(LLM-DDP) with highlignted five key components. Equations are defined in Section 3.4, 3.5 and 3.6.

closely mimic those of the expert. The formulation is as follows:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(s,a)\sim\mathcal{B}} \left[ L(\pi_\theta(s), a) \right]. \quad (1)$$

Here, $\mathcal{B}$ denotes the dataset comprising state action pairs (s, a). The loss function $L(\pi_\theta(s), a)$ quantifies the discrepancy between the action $\pi_\theta(s)$ produced by the policy $\pi$ in the state $s$ and the action of the expert $a$. The symbol $\theta^*$ represents the optimal set of parameters, which we aim to obtain through training for the policy network. Through the pre-training process, we attained an Inform score of 48.9 (dialogue information provision effectiveness), a Complete score of 42 (predefined task accomplishment degree), a Success score of 26.7 (dialogue objective fulfillment rate), and the count of Successful Turns reached 7.48 (dialogue turns for goal achievement). The results show that BC has effectively grasped the policy network, greatly speeding up the training process. This progress lays a good foundation for further reinforcement learning work.

### 3.3 Action probability sampling

In the subsequent reinforcement learning phase, this study uses the Bernoulli distribution to determine the action $A_t \in A$ to be taken. The Bernoulli distribution is a discrete probability distribution that models a random experiment with only two possible outcomes: success (denoted as 1) and failure (denoted as 0). Similarly, our action distribution is binary, consisting of only two values, 1 and 0. Let the probability of success be denoted by $p$ where $0 \leq p \leq 1$, then the probability of failure is $1 - p$. The Bernoulli distribution, denoted as $D(x)$, is specifically expressed as:

$$A_t = D(p(\hat{a}_t)) = p^x (1-p)^{1-x}. \quad (2)$$

Here, $p(\hat{a}_t)$ represents the probability of the action taken at time $t$ as predicted by the policy network. Given that we are dealing with discrete actions, the interaction between the agent and the environment is iterative. Each step relies on the current state and the agent's policy. This interaction is modeled as a Markov chain, where the state transition $P(S_{t+1} = s' \mid S_t = s, A_t = a)$ and the reward $R_{t+1} = R(S_t, A_t)$ are determined by the dynamics of the environment and the agent's policy. The agent aims to learn an optimal policy to maximize the cumulative long-term rewards.

### 3.4 Domain Priority Ranking

By integrating the dialogue context state $S_t$ set into the LLM prompts and leveraging the LLM's common sense knowledge to extract key information, we can ensure a comprehensive understanding of the dialogue content within the limitations of context length. This process enables the generation of probabilities for relevant domains, which are

expressed as follows:

$$p_{\text{LLM}}(\hat{a}_t) = \text{LLM}([S_t; prompt]). \quad (3)$$

Based on domain priority, the possible action space is filtered out to generate action masks. Then, perform an element-wise multiplication of the action masks with the output of the action by the policy network. This operation facilitates the integration of the two predictions. The expression is given by:

$$p_{\text{C}}(\hat{a}_t) = p_{\text{LLM}}(\hat{a}_t) \times A_t. \quad (4)$$

The fused action prediction probabilities are then passed through the Bernoulli distribution to determine the actions taken. The expression is as follows:

$$A_t(\text{domain}) = D(p_{\text{C}}(\hat{a}_t)) = p_{\text{C}}^x(1-p_{\text{C}})^{1-x}. \quad (5)$$

The LLM-DDP prompt is placed in the Appendix Table 3.

### 3.5 Multi Cross entropy loss function

This study employs an innovative policy network optimization method, which adjusts the policy network by minimizing the multi-cross-entropy loss between the fused action $A_t(domain)$ and the output of the action of the original policy network $A_t$. The multi-cross-entropy loss function is utilized to predict the difference in probability distributions between $A(domain)^{(i)}$ and $p(\hat{a}_t)^{(i)}$ at time step $t$. The expression is as follows:

$$L_{\text{CE}} = -\sum_{i=1}^{N} A(domain)^{(i)} \log(p(\hat{a}_t)^{(i)}). \quad (6)$$

Here, $A(domain)^{(i)}$ serving as the target action, can be regarded as a one-hot encoded label [0, 1]. $N$ denotes the numerical encoding of all actions that the system can take during execution. In the MultiWOZ 2.1 dataset, the value of N is 208. We use the multi-cross-entropy loss function to measure the divergence between the fused and original action distributions. Through iterative optimization of this loss, our goal is to guide the policy network to converge to an action-selection policy for handling domain-priority tasks.

### 3.6 Domain-driven loss function for critics

This study uses MSE, which is widely recognized for its applicability and reliability in regression tasks, as the loss function for the critic network. MSE provides a direct measure of model performance by quantifying the discrepancy between the predicted values of the value function and the target values. Furthermore, for the purpose of network optimization and the achievement of a balance between bias and variance, the generalized advantage estimate (GAE) is selected as the target value. GAE estimates the value function by combining temporal difference (TD) errors in multiple time steps, and this is done while achieving that bias-variance balance through the appropriate selection of the hyperparameter $\lambda$. The expression for GAE is as follows:

$$\begin{aligned} \text{GAE}(\tau) &= \sum_{k=0}^{\infty} \gamma^k \lambda^k \Delta V \\ \Delta V &= R_{t+k+1} + \gamma V(s_{t+k+1}) - V(s_{t+k}). \end{aligned} \quad (7)$$

Thus, the calculation of the MSE loss $L_{\text{MSE}}$ is given by:

$$L_{\text{MSE}} = \frac{1}{2} \sum_{t=\tau}^{N} \left(\text{GAE}(\tau) - Q(s_t, a_t)\right)^2. \quad (8)$$

As yet, the current loss function does not incorporate information on domain-priority actions. This limitation hampers the model's ability to capture specific domain characteristics. To solve this limitation, our study proposes a novel loss function design that aims to perform backpropagation twice, once with the MSE loss function and once with the multi-cross-entropy loss function. The formulation is as follows:

$$L_{\text{C}} = L_{\text{MSE}} + \lambda * L_{\text{CE}}. \quad (9)$$

The loss function $L_{\text{C}}$ combines the stability of MSE with the domain sensitivity of multi-cross-entropy. This integrated strategy improves the model's adaptability to the features of composite domains.

## 4 Experiment

### 4.1 Dataset

This study evaluates performance using the MultiWOZ 2.1 (Eric et al., 2020) benchmark data set. MultiWOZ 2.1 is an extensive Task-Oriented Dialogue dataset encompassing 10,425 dialogues across 7 distinct domains. There were 3,406 single-domain dialogues and 7,032 multi-domain dialogues. In addition, human evaluation was included

to accurately gauge algorithm performance. Since the MultiWOZ dataset is based on the MIT open-source license, it does not involve privacy-related issues or potential malicious or unintended harmful effects.

### 4.2 Experimental Setting

**Implementation** The experiments are conducted on a Linux server with 64 GB memory, Ultra9 CPU (24 cores), NVIDIA A6000 GPUs (48 GB). We implement the algorithm on the basis of ConvLab-3 (Zhu et al., 2022). The backend for deep learning is PyTorch (Paszke et al., 2019).

**State-action space** In MultiWOZ 2.1, the state-action space is defined by 361 dimensions and the action space is 208 dimensions.

**LLM model** Utilize the API interface of GPT-3.5-turbo to achieve the prediction of domain priority.

**Hyper-parameters for our LLM-DDP algorithm** The policy network is a 3-layer Multi-Layer Perceptron (MLP) model, with hidden size 512, and the ReLU activation at each layer. The critic network has the same architecture as the policy network.

In the training process, the seed was set to 42 for initializing random parameters. The discount factor gamma was set to 0.99 to calculate the discounted sum of future rewards. The lambda parameter in GAE was set to 0.95 to balance bias and variance. The clip ratio for the PPO loss (Wang et al., 2020b) was set to 0.2 to limit the magnitude of policy updates. The learning rate for the policy network was set to $3e{-}4$, determining the step size of parameter updates in the policy network. The learning rate for the value network was also set to $3e{-}4$, determining the step size of parameter updates in the value network. The maximum norm for gradient clipping was set to 0.5 to prevent gradient explosion. The number of iterations for training the policy network was set at 200 and the same number of iterations was used to train the value network.

### 4.3 Baselines

To systematically assess the effectiveness of the dialogue system approach proposed in this study, we consider five alternative methods, denoted SAi ($i = 1, 2, ..., 5$). Methods SA1 - SA4 were based on a pipeline architecture, utilizing NLU, DST, and NLG modules identical to those selected in this study, namely BERT, Rule, and Template, respectively. Method SA5 involved invoking the GPT-3.5-turbo API and using ChatGPT as an end-to-end system role. Further details are as follows:

**SA1 (DP Module with GDPL)** The DP module is adopted by the Guided Dialogue Policy Learning (GDPL) algorithm (Takanobu et al., 2019).

**SA2 (DP Module with PG)** The DP module is adopted by the Policy Gradient (PG) algorithm (Sutton et al., 1999).

**SA3 (DP Module with PPO)** The DP module is adopted by the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017).

**SA4 (DP Module with DQN)** The DP module is adopted by the Deep Q-Learning (DQN) algorithm (Hester et al., 2018).

**SA5 (End-to-End Dialogue Model with GPT3.5-turbo)** Specifically designed prompts are used to invoke the GPT-3.5-turbo interface to implement an end-to-end dialogue model. The prompt results are shown in the appendix table 4.

### 4.4 Evaluation Metrics

Following previous research (Jang et al., 2022; Peng et al., 2021, 2017; Wang et al., 2022), we evaluated our system using four metrics: Inform, Complete, Success, and Dialogue Turns. The Inform metric evaluates whether the dialogue system can accurately provide the required entity information and key content for the task. The Complete metric assesses the extent to which the system provides comprehensive information to assist users in completing their tasks. The Success metric measures the system's ability to successfully help users accomplish their predetermined tasks. Dialogue Turns refers to the number of conversational exchanges needed to complete a task, denoted as Turn(succ). Ideally, a lower number of dialogue turns indicates greater system efficiency. However, it may also suggest that no effective dialogue has taken place.

### 4.5 Main results

#### 4.5.1 Automatic Evaluation

In this study, we use four metrics, Inform, Complete, Success, and Turn (succ) to comprehensively evaluate the performance of our LLM-DDP model. To ensure the fairness and validity of the experiments, given that ConvLab-3 does not disclose its parameters, all baseline methods are standardized to match the parameters and training epochs of our LLM-DDP model. The results are presented in Table 1, which clearly indicates that LLM-DDP significantly outperforms other methods.

Among the experiments, SA4 and SA5 exhibit significantly lower metric values. The stark con-

| Method | Inform | Complete | Success | Turn (succ) |
|---|---|---|---|---|
| (SA1) DP Module with GDPL | **0.54** | 0.24 | 0.10 | 4.81 |
| (SA2) DP Module with PG | **0.54** | 0.45 | 0.11 | 5.81 |
| (SA3) DP Module with PPO | 0.51 | 0.63 | 0.28 | 6.35 |
| (SA4) DP Module with DQN | 0.03 | 0.17 | 0.01 | 19.8 |
| (SA5) End-to-End Dialogue Model with GPT3.5: | - | 0.24 | 0.08 | 9.69 |
| **LLM-DDP** | **0.54** | **0.86** | **0.45** | **15.05** |

Table 1: A Comparative Results Table Based on Uniform Experimental Parameters and Training Epochs

trast implies that the DQN algorithm and the approach that rely solely on prior knowledge of LLMs and prompts struggle to handle the complex dependencies in the composite domain scenario.

When looking at the Inform metric, SA1, SA2, and LLM-DDP share the same value of 0.54. This indicates that these three algorithms are comparable to provide required entities and key content. The PPO algorithm in SA3 shows a relatively better performance. LLM-DDP, which improves the PPO algorithm by integrating LLM with an innovative loss function, shows remarkable improvements. The success rate of LLM-DDP increases to 0.45, which is 59.7% higher than in SA3. This significant increase highlights the effectiveness of integrating LLMs and the novel loss function in enhancing the dialogue completion rate and the overall success of the system.

In the MultiWOZ 2.1 dataset, the average turns are 14. LLM-DDP has a Turn(succ) of 15.05, which is close to the average, while maintaining a high Complete metric of 0.86. This implies that LLM-DDP can complete dialogues efficiently, while other methods fail to reach the same level of performance.

In summary, whether considering other pipeline methods or end-to-end approaches, all perform inferiorly to LLM-DDP under the premise of unified parameters and training epochs. The experimental results strongly substantiate the effectiveness of our LLM-DDP approach in handling composite domain tasks in TOD systems.

#### 4.5.2 Human Evaluation

To enhance the accuracy of our evaluation, we enlisted human evaluators to assess dialogues. For the fairness and effectiveness of the evaluation, 25 human volunteers, including researchers and ordinary users, were selected. Each assessor conducted five conversations with six baseline methods and our proposed method. It was clearly stated to the assessors that the assessment data were sourced from open source datasets and did not involve privacy - related issues. The performance of the models was evaluated across four crucial dimensions: Content, Accuracy, Satisfaction, and Success.

In these metrics, higher scores mean better performance. Ratings are divided into the following four levels: First, when the user action perfectly aligns with the system action, a score of 100 is awarded. Second, when the user action largely matches the system action, the score is assigned based on the specific circumstances within the range of 50-99. Third, when the user action partially matches the system action, the score is assigned based on the specific circumstances within the range of 1-49. Fourth, when the user action is entirely mismatched with the system action, a score of 0 is given.

The manual grading results are presented in Figure 2 in Appendix E. According to Figure 2. Among SA1 to SA5, SA2 achieved the highest scores in terms of content, but was surpassed by SA3 in other metrics. The Accuracy, Satisfaction, and Success metrics of SA3 are higher than those of all other baselines. However, these metrics were consistently lower than those of LLM-DDP. LLM-DDP maintained robust performance across the human evaluation criteria.

### 4.6 Ablation studies and further analysis

**Ablations on our LLM-DDP framework** To comprehensively explore the influence of different techniques on the final experimental results, we conducted seven groups of ablation experiments in four major directions, as presented in Table 2.

- **NOMASK**: The LLM was not used to predict domain priority during training, so action probabilities related to the domain were not generated. When comparing LLM-DDP, all indicators of NOMASK are lower.This clearly demonstrates the significance of the domain priority algorithm in dealing with composite

| Method | Inform | Complete | Success | Turns (succ) |
|---|---|---|---|---|
| NOMASK | 0.52 | 0.81 | 0.42 | 13.17 |
| GC-0.1 | **0.55** | 0.82 | 0.41 | 15.05 |
| GC-0.3 | 0.53 | 0.71 | 0.40 | 14.34 |
| MSE-Huber | 0.52 | 0.85 | 0.40 | 16.52 |
| MSML-CEL | 0.52 | 0.81 | 0.42 | 13.5 |
| DM-50 | 0.51 | 0.71 | 0.41 | 5.82 |
| DM-125 | 0.50 | 0.79 | 0.42 | 8.82 |
| **LLM-DDP** | 0.54 | **0.86** | **0.45** | 15.05 |

Table 2: Comparison of the experimental results of the seven ablation experiments with those of the LLM-DDP experiments

domain scenarios. By predicting domain priority, LLM-DDP can better filter the action space and make more appropriate decisions, thereby improving the performance of the dialogue system.

- **GC-0.1** and **GC-0.3**: Represent that the PPO clipping ratios are 0.1 and 0.3, while LLM-DDP is 0.2. GC-0.1 achieved the best result on the Inform metric. Limiting the ratio values enables the model to provide more accurate information during the dialogue process. The values of the Complete and Success metrics are lower than those of LLM - DDP. This indicates that setting the hyperparameter of the clipping coefficient is crucial and has a relatively significant impact on the results.

- **MSE-Huber** and **MSML-CE**: MSE-Huber indicates replacing the MSE Loss with the Huber Loss, and MSML-CE means replacing Multi Label Soft Margin Loss with Cross Entropy Loss. The experimental results show that replacing the loss function has some impact on the performance of the dialogue system. However, this impact is relatively minor, indicating that the LLM-DDP algorithm exhibits strong robustness.

- **DM-50** and **DM-125**: Represent training the model for 50 and 125 epochs, respectively, while LLM-DDP is trained for 600 epochs. All indicators gradually increase from DM-50 to DM-125 and then to LLM-DDP. This indicates that the LLM-DDP algorithm requires a certain number of epochs to converge. But even with only 50 training epochs, the DM-50 results are still better than the indicators of the SA1-SA5 methods, demonstrating the superiority of the LLM-DDP framework.

These ablation experiments provide in-depth insight into the importance of each component and parameter in the LLM-DDP framework, further validating the effectiveness and rationality of the proposed method.

## 5 Conclusion

This study introduces the LLM-DDP framework, integrating LLM with domain-driven critic loss functions to resolve domain-dependent issues in composite domain tasks. With common sense knowledge and semantic comprehension of LLM, it significantly enhances the adaptability of the model to domain-related problems. The combined loss function design increases the efficiency and performance of the model. Comprehensive experiments, including automatic evaluations and human assessments on the MultiWOZ 2.1 dataset, have validated the superiority of our method. The results reinforce the effectiveness of the LLM-DDP framework.

## Limitations

However, we have yet to implement LLM-DDP in more LLM. To further enhance model performance, we plan to conduct experiments on higher-tier LLMs, especially reasoning models, in the future. In addition, we will expand our experimental scope to include richer datasets to ensure the generalizability of the model. Our exploration will also extend to more complex prompts.

## Ethical considerations

Our work strictly adheres to the ethical guidelines and principles outlined by the ACL. All data sets used in our research are sourced from previous studies, ensuring that there are no privacy concerns or issues related to racial discrimination.

## References


Vevake Balaraman and Bernardo Magnini. 2021. Domain-aware dialogue state tracker for multi-domain dialogue systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:866–873.

Vevake Balaraman, Seyedmostafa Sheikhalishahi, and Bernardo Magnini. 2021. Recent neural methods on dialogue state tracking for task-oriented dialogue systems: A survey. In *Proceedings of the 22nd annual meeting of the special interest group on discourse and dialogue*, pages 239–251.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.

Jun Gao, Liuyu Xiang, Huijia Wu, Han Zhao, Yiqi Tong, and Zhaofeng He. 2023. An adaptive prompt generation framework for task-oriented dialogue system. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1078–1089.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. 2018. Deep q-learning from demonstrations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Youngsoo Jang, Jongmin Lee, and Kee-Eung Kim. 2022. Gpt-critic: Offline reinforcement learning for end-to-end task-oriented dialogue systems. In *International Conference on Learning Representations*.

Lukasz Kaiser, Aidan N. Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. 2017. One model to learn them all. *CoRR*, abs/1706.05137.

Wai-Chung Kwan, Huimin Wang, Hongru Wang, Zezhong Wang, Bin Liang, Xian Wu, Yefeng Zheng, and Kam-Fai Wong. 2024. JoTR: A joint transformer and reinforcement learning framework for dialogue policy learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9578–9588, Torino, Italia. ELRA and ICCL.

Yohan Lee. 2021. Improving end-to-end task-oriented dialog system with a simple auxiliary task. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1296–1303.

Jionghao Lin, Wei Tan, Ngoc Dang Nguyen, David Lang, Lan Du, Wray Buntine, Richard Beare, Guanliang Chen, and Dragan Gašević. 2023. Robust educational dialogue act classifiers with low-resource and imbalanced datasets. In *International Conference on Artificial Intelligence in Education*, pages 114–125. Springer.

Andrea Madotto, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Language models as few-shot learner for task-oriented dialogue systems. *CoRR*, abs/2008.06239.

Paramita Mirza, Viju Sudhi, Soumya Ranjan Sahoo, and Sinchana Ramakanth Bhat. 2024. ILLUMINER: Instruction-tuned large language models as few-shot intent classifier and slot filler. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8639–8651, Torino, Italia. ELRA and ICCL.

Atsumoto Ohashi and Ryuichiro Higashinaka. 2022a. Adaptive natural language generation for task-oriented dialogue via reinforcement learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 242–252, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Atsumoto Ohashi and Ryuichiro Higashinaka. 2022b. Post-processing networks: Method for optimizing pipeline task-oriented dialogue systems using reinforcement learning. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–13, Edinburgh, UK. Association for Computational Linguistics.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. Soloist: Buildingtask bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2231–2240, Copenhagen, Denmark. Association for Computational Linguistics.

Mahdin Rohmatillah and Jen-Tzung Chien. 2023. Hierarchical reinforcement learning with guidance for multi-domain dialogue policy. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:748–761.

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

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 100–110, Hong Kong, China. Association for Computational Linguistics.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. 2020a. Modelling hierarchical structure between dialogue policy and natural language generator with option framework for task-oriented dialogue system. *CoRR*, abs/2006.06814.

Weizhi Wang, Zhirui Zhang, Junliang Guo, Yinpei Dai, Boxing Chen, and Weihua Luo. 2022. Task-oriented dialogue system as natural language generation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 2698–2703.

Yuhui Wang, Hao He, and Xiaoyang Tan. 2020b. Truly proximal policy optimization. In *Uncertainty in artificial intelligence*, pages 113–122. PMLR.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

Qingyang Wu, James Gung, Raphael Shu, and Yi Zhang. 2023. DiactTOD: Learning generalizable latent dialogue acts for controllable task-oriented dialogue systems. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 255–267, Prague, Czechia. Association for Computational Linguistics.

Weijie Xu, Zicheng Huang, Wenxiang Hu, Xi Fang, Rajesh Cherukuri, Naumaan Nayyar, Lorenzo Malandri, and Srinivasan Sengamedu. 2024. HR-MultiWOZ: A task oriented dialogue (TOD) dataset for HR LLM agent. In *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, pages 59–72, St. Julian's, Malta. Association for Computational Linguistics.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 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, pages 14230–14238.

Naoki Yoshimaru, Motoharu Okuma, Takamasa Iio, and Kenji Hatano. 2023. Asyncmld: Asynchronous multi-llm framework for dialogue recommendation system. *arXiv preprint arXiv:2312.13925*.

Yangyang Zhao, Mehdi Dastani, and Shihan Wang. 2024. Bootstrapped policy learning: Goal shaping for efficient task-oriented dialogue policy learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 2615–2617.

Qi Zhu, Christian Geishauser, Hsien-chin Lin, Carel van Niekerk, Baolin Peng, Zheng Zhang, Michael Heck, Nurul Lubis, Dazhen Wan, Xiaochen Zhu, et al. 2022. Convlab-3: A flexible dialogue system toolkit based on a unified data format. *arXiv preprint arXiv:2211.17148*.

Ying Zhu, Yameng Li, Yuan Cui, Tianbao Zhang, Daling Wang, Yifei Zhang, and Shi Feng. 2023. A knowledge-enhanced hierarchical reinforcement learning-based dialogue system for automatic disease diagnosis. *Electronics*, 12(24):4896.

# A Preliminaries

## A.1 Pipeline architecture for TOD

The pipeline architecture is a classic design paradigm for TOD systems, which decomposes the entire system into multiple modules. Typically, it consists of four core modules: NLU, DST, DP and NLG, each responsible for a specific subtask (Ohashi and Higashinaka, 2022b).

## A.2 NLU module for TOD

A common approach for NLU involves training a BIO (Begin, Inside, Outside) tagger for slot-value pairs and a multiclass classifier for intents. The slot-filling task takes the user's utterance $X$ as input and generates a dictionary $M = \{s_1 = v_1, \ldots, s_n = v_n\}$, where $s_1$ represents a slot and $v_i$ represents a corresponding value for that slot (Madotto et al., 2020).

## A.3 DST module for TOD

Given a dialogue $D$ consisting of $t$ turns of utterances $X_U^1, X_S^1, \ldots, X_U^t$, a DST model predicts a dictionary $M_t = \{s_1 = v_1, \ldots, s_n = v_n\}$, similar to the process of natural language understanding (NLU).

## A.4 DP module for TOD

The DP module determines the next action of the system by integrating the current belief state $M_t$, the historical context, and the results of the database query (Wang et al., 2022). Specifically, it

combines the current understanding of the user's intent with contextual information and selects an appropriate response from the database.

### A.5 NLG module for TOD

The Natural Language Generation (NLG) module is tasked with the responsibility of translating the system's decisions or actions into natural and fluent language. The model takes a speech act and a slot-value dictionary as input and generates a discourse $X$ as output.

## B Prompt for LLM-DDP

For the LLM-DDP, we designed the corresponding prompts and obtained the respective answers, as shown in the Appendix Table 3.

## C Prompt for SA5 Experiments

For the SA5 experiment, we designed the corresponding prompts and obtained the respective responses, as shown in the appendix Table 4.

## D Pseudocode of the PPO algorithm

To better explain the PPO algorithm, we present its pseudocode, as shown in Table 5.

## E Human Grading

In Figure 2, the Content metric assesses whether the system provided the correct actions and feedback in response to user requests. The Accuracy metric evaluates whether users received precise answers through brief dialogues, including instances where the system offered responses to unasked questions. The Satisfaction metric reflects overall user contentment with the system's responses and performance over five interactions. The Success rate indicates whether the system ultimately met all requirements. Users are required to rate the model based on these four criteria.

| **Prompt for LLM-DDP** |
|---|
| In a task-oriented dialogue scenario, you will be provided with user actions from historical dialogues presented in JSON format. Analyze the patterns and information within these historical user actions. Consider the interdependent relationships among different domains. Think about how different actions might be associated with specific domains. Based on this analysis, predict the possible domains to which the current dialogue belongs. The output should be in JSON format, including the possible domains and their corresponding probability values.<br>user actions: [["inform", "train", "NotBook", "none"], ["inform", "train", "leave at", "18:30"], ["inform", "train", "departure", "cambridge"], ["inform", "train", "day", "wednesday"], ["inform", "train", "arrive by", "dontcare"], ["inform", "train", "destination", "stevenage"]] |
| **Answer** |
| ["hotel":0.3, "train":0.8] |

Table 3: LLM-DDP Prompt in the Appendix

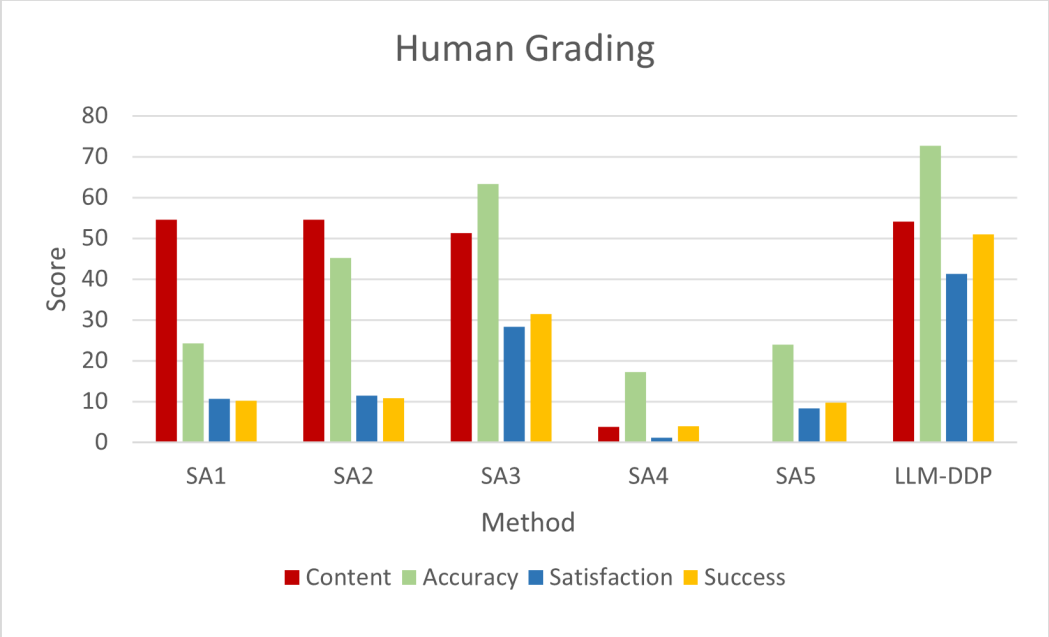


Figure 2: The distribution chart of manual grading for the six methods.

| **Prompt for SA5 Experiments** |
|---|
| {'role': 'system','content': 'Imagine you are a user chatting with a helpful assistant to achieve a goal. You should chat according to the given goal faithfully and naturally. You should not generate all the information in the goal at once. You should generate short, precise, and informative response (less than 50 tokens), corresponding to only one or two items in the goal. You should not generate information not presented in the goal. If and only if you achieve your goal, express your thanks and generate **"[END]"** token. If you think the assistant can not help you or the conversation falls into a infinite loop, generate **"[STOP]"** token.'}, {'role': 'user', 'content': 'I need a hotel please. I need one with wifi. Are there any 4 stars available? It must have free parking.'}, {'role': 'assistant', 'content': 'Sure, I can help you with that. I found a 4 - star hotel with free parking and wifi. Would you like me to provide more details?'}, {'role': 'user', 'content': 'I would like it to have a 4 star rating.'} |
| **Answer** |
| I have found a 4 - star hotel with free parking and wifi. Would you like me to proceed with booking this hotel for you? |

Table 4: SA5 Experiments Prompt in the Appendix

| **Pseudocode of the LLM-PPO algorithm** |
|---|
| **for** iteration = 1, 2, ... **do**<br>  **for** actor = 1, 2, ..., N **do**<br>  Run policy $\pi_{\theta_{\text{old}}}$ in environment for $T$ timesteps<br>  Using LLM to generate the probability distribution $P_a$ in the action domain<br>  Compute action domain Cross Entropy Loss<br>  Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$<br>  **end for**<br>  Compute Mean Square Error Loss<br>  Optimize surrogate $L$ wrt $\theta$, with $K$ epochs and minibatch size $M \leq NT$<br>  $\theta_{\text{old}} \leftarrow \theta$<br>**end for** |

Table 5: Pseudocode in the Appendix