Semantic-Aware Action Space Compression via LLM-DRL Synergy for Efficient Task-oriented Dialogue Policy Exploration

Anonymous ACL submission

Abstract

The flexibility of natural language significantly expands the action space in task-oriented dialogue systems, causing inefficient exploration and slow convergence in deep reinforcement learning (DRL)-based policy optimization. Pretrained large language models (LLMs), with world knowledge and semantic understanding, offer promising solutions. To this end, we propose LLM-Guided DRL via Semantic-Aware Action Pruning (LLMSAP), a novel framework that synergizes pretrained LLMs with DRL. LLMSAP leverages the world knowledge and contextual understanding of LLMs to guide decision-making via an action feasibility assessment. Instead of requiring LLMs to directly generate optimal actions due to their limited precision in sequential decision tasks, LLMSAP employs a lightweight action pruning mechanism. Specifically, LLMs act as action filters, rapidly eliminating semantically implausible or low-potential actions from multi-turn dialogue context, allowing the DRL agent to focus exploration on a refined candidate subset. This two-stage framework ("prune-then-optimize") avoids extensive LLM fine-tuning while preserving the decision-making precision of DRL. Experiments on multiple benchmarks verify the effectiveness of LLMSAP.

1 Introduction

Task-oriented dialogue (TOD) systems achieve user goals via multi-turn interactions, where dialogue policy (DP), as the core component, selects appropriate actions to steer the conversation. Deep reinforcement learning (DRL) has emerged as the dominant approach for DP optimization due to its strength in sequential decision-making (Du et al., 2024). However, natural language flexibility induces exponentially large action spaces (Zhao et al., 2024a). The resulting high dimensionality and semantic ambiguities in actions not only hinder efficient exploration (Ma et al., 2024) but also destabilize policy optimization through biased value estimation (Zhao et al., 2019), degrading DRL's generalization in complex TOD scenarios.

Recent advances in large language models (LLMs), which acquire powerful world knowledge and multi-turn semantic awareness through massive corpus pretraining, offer promising solutions to the above challenges (Chung et al., 2023). Studies show that LLMs could deeply model implicit semantic relationships in dialogue contexts and infer relevant system action sets accordingly (Qian et al., 2024). This capability offers a theoretical foundation for developing semantic-aware action pruning modules, enabling semantically-guided DRL exploration in expansive action spaces.

Despite their potential, deploying LLMs as end-to-end decision-makers for dialogue policy guidance faces dual challenges: (1) Untuned LLMs lack alignment with TOD-specific reward signals, making it difficult to optimize long-term action rewards (Algherairy and Ahmed, 2025); 2) Large-scale fine-tuning for specific tasks is hindered by the scarcity of annotated TOD data and high computational costs (Matarazzo and Torlone, 2025).

To bridge these gaps, we propose LLM-Guided DRL via Semantic-Aware Action Pruning (LLMSAP), a two-stage "prune-then-optimize" framework. It first leverages LLMs to assess action feasibility based on dialogue context, pruning semantically inconsistent or low-potential actions to compress the action space. Then, DRL performs fine-grained exploration over the pruned action subset, optimizing action sequences via long-term reward maximization. This design avoids costly LLM fine-tuning while harnessing its role as a *semantic filter*, balancing DRL policy optimization accuracy with exploration efficiency. To the best of our knowledge, this is the first study to integrate LLM with DRL to dialogue policy optimization. In summary, our contributions are threefold:

- A lightweight semantic action pruning mechanism that leverages LLMs filtering to com-

press the action space efficiently, effectively alleviating the exploration bottleneck of DRL in high-dimensional environments;

- A cross-modal decision fusion framework that unites LLM-derived semantic insights with DRL policy gradients through prompt-based action feasibility evaluation;
- Experiments on multiple benchmarks demonstrate that LLMSAP accelerates convergence and boosts task completion rates, showcasing the synergistic benefits of combining semantic guidance with deep reinforcement learning.

2 Related Work

In DRL research, LLMs typically serve as information processors, reward designers, decision-makers, or generators (Cao et al., 2024). Nevertheless, existing studies predominantly concentrate on gaming environments, exhibiting scant exploration of TOD systems. Due to fundamental differences in task specifications (discrete vs. continuous action spaces) and interaction patterns (turn-based vs. game dynamics), gaming approaches are ill-suited for TOD. This paper investigates analogous LLM-as-decision-maker methods in related fields, categorizing them into action decision-making and action guidance paradigms, to inform our exploration of LLM-DRL integration for TOD.

For decision-making, recent advances have explored the use of LLMs through two main paradigms: policy initialization and sequence modeling-based decision making. The policy initialization paradigm utilizes pre-trained LLMs to provide strong priors for DRL (Li et al., 2022). In contrast, the sequence modeling paradigm reframes decision-making as a conditional generation problem, typically implemented via decision transformers (Shi et al., 2023). However, these approaches share a fundamental limitation: dialogue policy learning is a long-term task, and LLMs without task-specific fine-tuning often fail to generate an optimal sequence of actions (Yi et al., 2024b).

For action-guiding, LLMs do not generate actions directly, but act as guides, producing a condensed set of candidate or expert actions. Hu and Sadigh (2023) proposed the *instructRL* framework, which utilizes pre-trained LLMs to generate a priori strategy distributions based on linguistic cues to guide policy learning. However, *instructRL* relies on LLMs to generate strategy distributions directly,

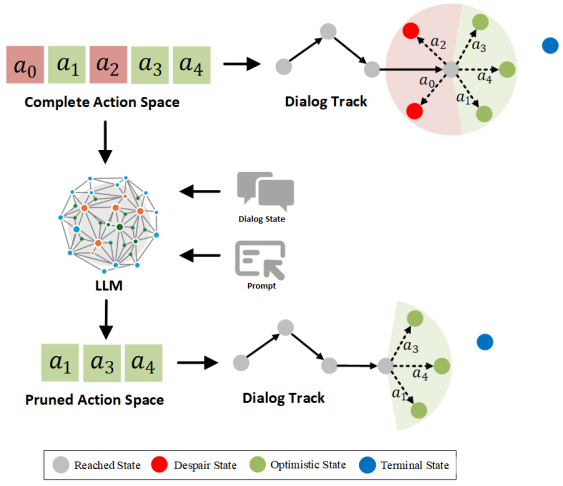


Figure 1: The first stage of LLMSAP utilizes LLMs to prune redundant or invalid actions.

which has significant limitations in long sequence tasks due to cumulative bias propagation and failure to maintain temporal consistency in action sequences. Meanwhile, DRL agents are subordinate and lack the autonomy to correct policy deviations, which may lead to unstable convergence or even failure, especially in complex and dynamic dialogue tasks (Kwan et al., 2023).

In summary, direct policy initialization or action generation with LLMs suffers from limited adaptability and long-term optimization in TOD scenarios. In contrast, LLMSAP utilizes LLMs for semantic-level action pruning, avoiding their limitations in long-horizon decision-making while retaining the strengths of DRL, thus significantly enhancing efficiency in complex dialogue tasks.

3 Methodology

LLMSAP comprises two stages: (1) *LLM-Driven Semantic Action Pruning*: Leveraging the powerful semantic reasoning and world knowledge of pre-trained LLMs, this stage filters out semantically inconsistent or low-potential actions, thereby significantly reducing the effective action space; (2) *DRL-Driven Dynamic Optimization*: Operating within the pruned action subset provided by the LLM, this stage employs DRL to fine-tune the dialogue policy via policy-gradient updates, ensuring precise and efficient decision-making.

3.1 LLM-Driven Semantic Action Pruning

As illustrated in Figure 1, our framework employs a prompt-based interaction mechanism with LLMs, where the current dialogue state and the full action

space are encoded in natural language and integrated into the prompt (detailed prompt design in Appendix H). Leveraging their world knowledge and semantic reasoning, LLMs perform context-aware semantic analysis of actions by incorporating the historical context. By eliminating redundant or invalid actions, the LLM output retains only those most relevant to the current dialogue state, thereby reconstructing a pruned action space. To facilitate seamless integration with the DP network, LLM outputs are formatted as action subscripts, and the pruned action space is returned as a JSON string. By applying semantic action pruning to the action space, LLMs mitigate interference from irrelevant actions, enabling the agent to focus on high-value candidates. This process enhances exploration efficiency and accelerates DP training convergence.

3.2 DRL-Driven Dynamic Optimization

Task-oriented dialogue policies operate within finite, discrete action spaces. The DQN algorithm is well-suited for this setting, offering stable training and efficient offline learning and prior work shows DQN consistently outperforms continuous-control algorithms like PPO (Schulman et al., 2017) and SAC (Haarnoja et al., 2018) on discrete decision tasks. Thus, this paper employs DQN to learn optimal policies within the LLM-pruned action set, quantifying how semantic action pruning impacts dialogue performance.[1]

DQN extends Q-learning to high-dimensional spaces by replacing the value table with a deep network $Q(s, a; \theta)$ that estimates the expected return of executing action a in state s. For a given state s_t, the network outputs Q-values for all $a \in A$, and the parameters θ are updated to maximize long-term reward. The training objective is to minimize the following mean squared error loss function:

$$L(\theta) = \mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{D}}\left[(y_t - Q(s_t, a_t; \theta))^2\right] \quad (1)$$

$$y_t = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-) \quad (2)$$

where $Q(s_t, a_t; \theta)$ represents the expected cumulative reward from taking action a_t in state s_t. $\mathcal{D}$ is the experience replay buffer, which stores state transition tuples (s_t, a_t, r_t, s_{t+1}). γ is the discount factor balancing immediate rewards and long-term returns. θ^- denotes the parameters of the target Q-network, periodically synchronized from the online network θ to stabilize the training process.

[1]The pruned action space remains finite, enabling substitution of DQN with any discrete-action RL algorithm.

Within the pruned action space, DQN follows an ϵ-greedy strategy for action selection, conditioned on the current dialogue state. At each step, the agent explores by randomly selecting an action from the pruned set with probability ϵ or exploits by choosing the highest Q-value action with probability $1 - \epsilon$. This process is detailed in Appendix I.

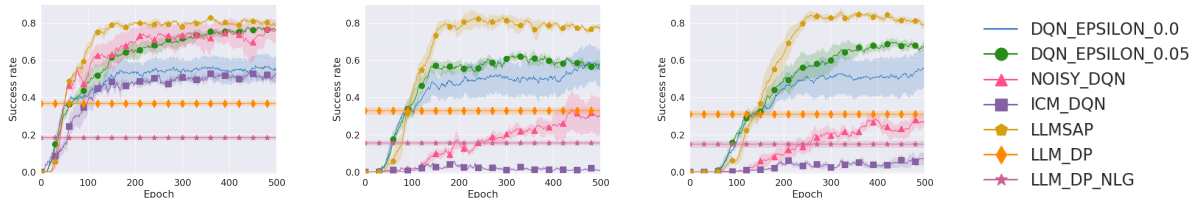
4 Experiments

We conducted experiments on three datasets from the Microsoft Dialogue Challenge platform (Li et al., 2018): movie ticket booking, restaurant reservation, and taxi booking. The objectives were to: (1) Demonstrate the superiority of LLMSAP in enhancing exploration efficiency (subsection 4.1); (2) Analyze the impact of LLM scale and compatibility on performance (subsection 4.2); (3) Investigate how semantic action pruning influences optimal exploration rate (subsection 4.3); (4) Validate effectiveness via human evaluation (subsection 4.4);

Given the focus on addressing exploration inefficiencies in expanded action spaces using LLMs, we selected baseline methods categorized into two groups: (1) Exploration-Enhanced DRL Policies: DQN_EPSILON_N (Mnih et al., 2015), NOISY_DQN (Han et al., 2022) and ICM_DQN (Lin et al., 2025); (2) LLM-Based Dialogue Policies: LLAMA_DP (Yi et al., 2024a) and LLAMA_DP_NLG (Yi et al., 2024a). Additional baselines and implementation details are provided in Appendix A and Appendix B.

4.1 Main Results

Figure 2 shows the learning curves of different agents across three datasets. NOISY_DQN and ICM_DQN performed well in the simple Movie task but degraded significantly in complex Restaurant/Taxi environments, due to inefficient exploration in large action spaces. This decline is mainly due to their inefficient exploration strategies that struggle to traverse the extensive action spaces characteristic of intricate dialogue scenarios. LLAMA_DP and its variants, on the other hand, leveraged world knowledge from pre-trained LLMs for strong initial performance, but lack of task fine-tuning hindered improvement. In contrast, LLMSAP uses LLM semantic understanding to prune actions and reduce redundant exploration, achieving faster convergence and higher success rates across all tasks. Detailed numerical results are in Appendix C.



(a) Movie (b) Restaurant (c) Taxi

Figure 2: The learning curves of different agents on three datasets. For the DQN_EPSILON_N, we set $N = 0$ to represent no exploration, while $N = 0.05$ denotes its performance upper bound (optimal exploration). The impact of different N values is detailed in Appendix G.

4.2 LLM Scale and Compatibility Analysis

To evaluate the impact of LLM scale, we conducted semantic action pruning utilizing LLMs of different sizes. As shown in Figure 3, llama-3.1-8B (Dubey et al., 2024) achieves performance comparable to the 70B model while significantly reducing computational costs. Therefore, we adopt llama-3.1-8B as the primary backbone for experiments, leveraging its balance of efficiency and effectiveness.

Moreover, our approach is theoretically compatible with any LLM. To validate this, we applied it to other mainstream models, including gemma-2-9B (Team et al., 2024) and qwen2.5-7B (Yang et al., 2024), which are similar in scale to llama-3.1-8B. Experimental results across three domains (see Appendix E) show that all three LLMs outperform baseline approaches. These results confirm the effectiveness of our approach and demonstrate its broad compatibility with various LLMs.

4.3 Impact of Semantic Action Pruning on Exploration Efficiency

We investigated how LLM-driven semantic action pruning affects the selection of the ϵ hyperparameter. By varying ϵ from 0.05 to 0.25, we discovered that $\epsilon = 0.15$ strikes the optimal balance between exploration and exploitation. Unlike DRL-based methods, which attained its peak performance at $\epsilon = 0.05$, the LLM-based approachs profits from a higher ϵ value. This is because it prunes redundant or invalid actions, thereby enabling more extensive yet productive exploration and enhancing training efficiency. Consequently, we set $\epsilon = 0.15$ for all subsequent experiments. The complete results and comparisons are presented in Appendix D.

4.4 Human Evaluation

While automated metrics offer quantitative assessments, human evaluation better reflects user experience by capturing task accuracy, dialogue coherence, and naturalness in multi-turn interactions. Human evaluations showed LLMSAP outperformed baselines across all dimensions, aligning with simulation results. The detailed settings and results are provided in Appendix F.

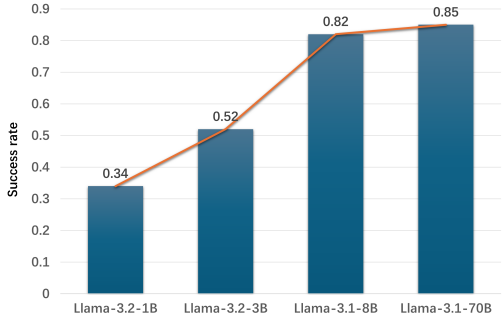


Figure 3: Performance on Llama with Different Scales.

5 Conclusion

This study introduces the LLMSAP architecture, a novel framework that synergizes LLMs with DRL to enhance exploration efficiency in task-oriented dialogue policy optimization. The framework initiates by conducting semantic interpretation of the ongoing dialogue context and action space, where an LLM is employed to eliminate redundant or semantically inconsistent actions, thereby generating a streamlined and high-purity action subset. DRL is subsequently utilized to execute exploration and policy refinement within this pruned action space. Cross-domain experiments spanning multiple scenarios reveal that LLMSAP surpasses standalone DRL and LLM methods in both exploration efficiency and convergence velocity. Its consistent performance across diverse LLM variants further underscores its robust generalizability. To the best of our knowledge, this is the first study to integrate LLM with DRL to dialogue policy optimization.

6 Limitations

Although the integration of LLMs and DRL in this study demonstrates clear advantages in improving exploration efficiency, the system requires LLMs to return data in a specific format. Any deviation from the expected structure may hinder accurate parsing and processing and thus compromise the system's accuracy and stability. To address this limitation, future research could explore fine-tuning LLMs to ensure consistent adherence to the required output format and improve the robustness and overall performance of the system.

References

Atheer Algherairy and Moataz Ahmed. 2025. Prompting large language models for user simulation in task-oriented dialogue systems. *Computer Speech & Language*, 89:101697.

Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. 2024. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems*.

Willy Chung, Samuel Cahyawijaya, Bryan Wilie, Holy Lovenia, and Pascale Fung. 2023. Instructtods: Large language models for end-to-end task-oriented dialogue systems. *arXiv preprint arXiv:2310.08885*.

Huifang Du, Shuqin Li, Minghao Wu, Xuejing Feng, Yuan-Fang Li, and Haofen Wang. 2024. Rewarding what matters: Step-by-step reinforcement learning for task-oriented dialogue. *arXiv preprint arXiv:2406.14457*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR.

Shuai Han, Wenbo Zhou, Jiayi Lu, Jing Liu, and Shuai Lü. 2022. NROWAN-DQN: A stable noisy network with noise reduction and online weight adjustment for exploration. *Expert Syst. Appl.*, 203:117343.

Hengyuan Hu and Dorsa Sadigh. 2023. Language instructed reinforcement learning for human-ai coordination. In *International Conference on Machine Learning*, pages 13584–13598. PMLR.

Wai-Chung Kwan, Hong-Ru Wang, Hui-Min Wang, and Kam-Fai Wong. 2023. A survey on recent advances and challenges in reinforcement learning methods for task-oriented dialogue policy learning. *Machine Intelligence Research*, 20(3):318–334.

Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. 2022. Pretrained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212.

Xiujun Li, Yu Wang, Siqi Sun, Sarah Panda, Jingjing Liu, and Jianfeng Gao. 2018. Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *arXiv preprint arXiv:1807.11125*.

Jie Lin, Yuhao Ye, Shaobo Li, Hanlin Zhang, and Peng Zhao. 2025. Improving exploration in deep reinforcement learning for incomplete information competition environments. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Sihong Liu, Jinchao Zhang, Keqing He, Weiran Xu, and Jie Zhou. 2021. Scheduled dialog policy learning: An automatic curriculum learning framework for task-oriented dialog system. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1091–1102.

Runyu Ma, Jelle Luijkx, Zlatan Ajanovic, and Jens Kober. 2024. Explorllm: Guiding exploration in reinforcement learning with large language models. *arXiv preprint arXiv:2403.09583*.

Andrea Matarazzo and Riccardo Torlone. 2025. A survey on large language models with some insights on their capabilities and limitations. *arXiv preprint arXiv:2501.04040*.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, et al. 2024. Tell me more! towards implicit user intention understanding of language model driven agents. arxiv [cs. cl](feb. *arXiv preprint arXiv:2402.09205*.

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

Ruizhe Shi, Yuyao Liu, Yanjie Ze, Simon S Du, and Huazhe Xu. 2023. Unleashing the power of pre-trained language models for offline reinforcement learning. *arXiv preprint arXiv:2310.20587*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. 2024a. A survey on recent advances in llm-based multi-turn dialogue systems. *CoRR*, abs/2402.18013.

Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. 2024b. A survey on recent advances in llm-based multi-turn dialogue systems. arxiv 2024. *arXiv preprint arXiv:2402.18013*.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. *arXiv preprint arXiv:1902.08858*.

Yang Zhao, Hua Qin, Zhenyu Wang, Changxi Zhu, and Shihan Wang. 2022. A versatile adaptive curriculum learning framework for task-oriented dialogue policy learning. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 711–723. Association for Computational Linguistics.

Yangyang Zhao, Mehdi Dastani, Jinchuan Long, Zhenyu Wang, and Shihan Wang. 2024a. Rescue conversations from dead-ends: Efficient exploration for task-oriented dialogue policy optimization. *Transactions of the Association for Computational Linguistics*, 12:1578–1596.

Yangyang Zhao, Ben Niu, Mehdi Dastani, and Shihan Wang. 2024b. Bootstrapped policy learning for task-oriented dialogue through goal shaping. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4566–4580.

A Baselines

We compared our approach with publicly available dialogue agents, dialogue agents designed to enhance exploration efficiency, and promising LLM-based dialogue agents:

- **DQN_EPSILON**_N agents are trained utilizing standard DQN with a traditional $\epsilon - greedy$ exploration strategy, where $\epsilon = N$ (Mnih et al., 2015)[2].
- **NOISY_DQN** agents enhance exploration by introducing noise into the network weights (Han et al., 2022).
- **ICM_DQN** agents incorporate intrinsic curiosity rewards to encourage exploration of the new space (Lin et al., 2025).
- **LLAMA_DP** agents replace the DP module of the TOD system with an LLM, selecting suitable actions to be passed to the NLG for response generation (Yi et al., 2024a)[3].
- **LLAMA_DP_NLG** agents replace both the DP and NLG modules of the TOD system with an LLM, directly selecting suitable words to construct responses (Yi et al., 2024a)[3].

To ensure a fair comparison, we directly employ pretrained LLMs to replace the DP or NLG modules, without involving any further fine-tuning on additional data. Therefore, our focus is placed on comparing the performance of LLAMA_DP and LLAMA_DP_NLG with that of our fully converged approach.

B Implementation Details

All DQN-based agents use a multilayer perceptron containing two hidden layers, each containing 80 neurons, with an activation function of ReLU. In the training configuration, the discount factor γ is set to 0.99, the batch size is 16, the learning rate is 0.001, and the experience replay buffer size is 5000. For the DQN_EPSILON_N agent, the ϵ starts at 0 and increases to 0.25 with a step size of 0.05. In the analysis of the impact of LLMs on the exploration rate, the ϵ ranges from 0.05 to 0.25, and the step size is 0.05. In our approach, epsilon is set to a default value of 0.15, as this setting yields the best performance. Detailed justification can be found in subsection 4.3. The default model scale for llama is set to 8B, as justified in subsection 4.2. The reward function assigns a reward of $2L$ for a successful dialogue and a penalty of $-L$ for a failed one. Additionally, to encourage concise conversations, a penalty of -1 is applied for each dialogue turn. All agents have a maximum dialogue turn limit of 30. Before the formal training begins, each agent undergoes 120 warm-up interactions with a rule-based user simulator to populate the experience replay buffer for subsequent training. During the training phase, each agent interacts with the environment once per episode and stores its experiences in the replay buffer. In the evaluation phase, each agent interacts with the environment 50 times, but its experiences are not stored in the buffer; instead, only the dialogue success rate, number of dialogue turns, and average reward are recorded. To ensure the robustness of experimental results, each agent is trained five times with different random seeds, and the average results are used for analysis.

C Main Result

The results of different agents across various domains are presented in Table 1. As shown in the table, the difference between epsilon values of 0 and 0.05 highlights the importance of exploration for the agent. Excessive exploration can lead to overly random strategies, preventing the agent from fully utilizing known information, and thus affecting convergence speed and stability. Conversely, insufficient exploration can cause the agent to get trapped in local optima, lacking the necessary diversity and flexibility to discover

[2]We assessed the impact of different ϵ values on performance across multiple dialogue domains, selecting the optimal value for each domain as the baseline. Detailed results and analysis are provided in Appendix G.

[3]We chose llama as the base model primarily due to its strong language understanding capabilities and open accessibility. In addition, we compare variants based on other LLMs to evaluate the generalizability of the proposed framework, as detailed in Appendix H.

the global optimum. Although NOISY_DQN demonstrates superior performance in the Movie domain, its performance significantly drops in the more complex state-action spaces of the Restaurant and Taxi domains. We hypothesize that, as the user objective becomes more complicated and the state space grows (Zhao et al., 2022), the role of the noise layer in facilitating exploration becomes limited, preventing the agent from effectively exploring more valuable policies. Similarly, the ICM_DQN agent suffers from this issue. While its convergence success rate reaches around 0.5 in the Movie domain, it drops below 0.1 in the Restaurant and Taxi domains. We believe that, in larger state spaces, ICM_DQN overly focuses on environmental dynamics rather than the task itself, leading to excessive, ineffective exploration. Although the LLAMA_DP and LLAMA_DP_NLG agents show impressive initial performance due to their built-in world knowledge, they fail to further improve performance due to the lack of task-specific data fine-tuning.

In contrast, LLMSAP benefits from the world knowledge and semantic understanding of LLMs, effectively eliminating redundant or invalid actions within the action space. This allows our approach to achieve the best performance across all three domains, with improvements in both convergence speed and task success rate, demonstrating that semantic action pruning of the action space by LLMs significantly enhances exploration efficiency.

Table 1: Results of different agents on three datasets, with top performance in each column highlighted. All results of agent pairs are statistically significant at the same epoch (t-test, $p < 0.05$). Epochs (50, 250, 500) represent early, mid, and post-convergence training stages.

Domain	Agent	Epoch = 50			Epoch = 250			Epoch = 500		
		Success↑	**Reward↑**	**Turns↓**	**Success↑**	**Reward↑**	**Turns↓**	**Success↑**	**Reward↑**	**Turns↓**
Movie	DQN_EPSILON_0.0	0.3505	-13.00	32.11	0.5403	12.99	25.70	0.5553	14.95	25.37
	DQN_EPSILON_0.05	0.3093	-18.61	33.44	0.6795	31.84	21.39	0.7668	43.42	19.21
	DQN_EPSILON_0.15	0.2086	-22.67	35.64	0.5137	11.61	28.18	0.5248	14.52	28.49
	NOISY_DQN	0.4137	-4.73	30.75	0.7141	36.68	20.04	0.7280	39.38	20.16
	ICM_DQN	0.1475	-37.81	33.00	0.5166	10.37	25.23	0.5311	12.49	24.47
	LLAMA_DP	**0.3845**	**-3.59**	**26.72**	0.3845	-3.59	26.72	0.3845	-3.59	26.72
	LLAMA_DP_NLG	0.1932	-26.73	28.31	0.1932	-26.73	28.31	0.1932	-26.73	28.31
	LLMSAP	0.3459	-13.90	32.83	**0.8081**	**48.49**	**18.96**	**0.8142**	**49.51**	**16.6**
Rest.	DQN_EPSILON_0.0	0.0695	-36.57	27.66	0.4907	4.10	22.13	0.5671	11.63	23.22
	DQN_EPSILON_0.05	0.0726	-36.28	27.63	0.5712	12.30	20.21	0.5817	12.79	21.12
	DQN_EPSILON_0.15	0.0348	-38.66	29.32	0.3443	-2.13	27.69	0.3016	-5.65	30.47
	NOISY_DQN	0.0000	-43.92	29.84	0.1669	-28.25	28.55	0.2988	-15.20	26.18
	ICM_DQN	0.0067	-40.85	24.90	0.0231	-38.92	23.99	0.0082	-32.88	**9.25**
	LLAMA_DP	**0.3464**	**-10.77**	**23.12**	0.3464	-10.77	23.12	0.3464	-10.77	23.12
	LLAMA_DP_NLG	0.1830	-28.44	35.60	0.1830	-28.44	35.60	0.1830	-28.44	35.60
	LLMSAP	0.0384	-53.36	35.96	**0.8163**	**49.64**	**18.63**	**0.7962**	**46.99**	18.32
Taxi	DQN_EPSILON_0.0	0.0004	-42.69	27.47	0.4846	2.26	24.70	0.5879	12.38	23.06
	DQN_EPSILON_0.05	0.0000	-42.86	27.71	0.5598	8.19	22.38	0.6683	20.19	21.90
	DQN_EPSILON_0.15	0.0009	-40.38	26.16	0.4186	1.13	26.56	0.4163	1.09	26.97
	NOISY_DQN	0.0000	-43.73	29.46	0.1455	-30.56	29.32	0.2615	-19.46	28.00
	ICM_DQN	0.0008	-42.34	26.84	0.0481	-34.48	19.62	0.0706	-28.59	**11.90**
	LLAMA_DP	**0.3288**	**-14.56**	**24.97**	0.3288	-14.56	24.97	0.3288	-14.56	24.97
	LLAMA_DP_NLG	0.1786	-18.33	28.46	0.1786	-18.33	28.46	0.1786	-18.33	28.46
	LLMSAP	0.0003	-43.47	35.02	**0.8220**	**48.62**	**19.44**	**0.8071**	**46.85**	19.23

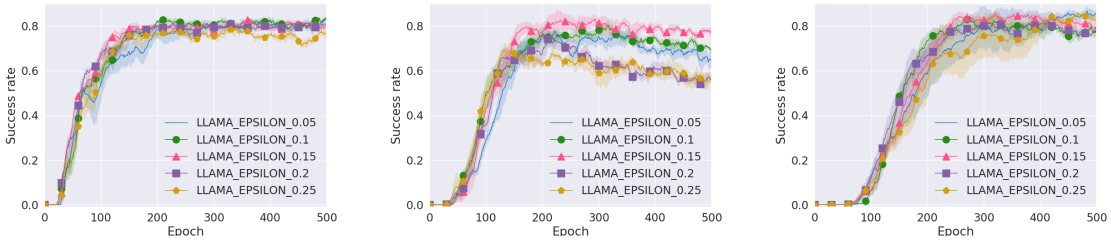
D Impact of LLMs Semantic Action Pruning on Exploration Rate

To examine the influence of LLMs' semantic action pruning of the action space on the ϵ hyperparameter and to provide valuable insights for future research and practical applications, we conducted a series of experiments on the ϵ hyperparameter. Intuitively, the magnitude of ϵ determines the extent of exploration. A larger ϵ increases the exploration frequency but does not exhibit a strictly linear relationship with exploration efficiency. An excessively large ϵ may lead to over-exploration, particularly in dialogue tasks with extensive state spaces, thereby degrading the quality of experiences. Conversely, an excessively small ϵ may result in insufficient exploration, causing the model to become trapped in local optima. Therefore, identifying an optimal ϵ value is essential for achieving an effective balance between exploration and exploitation.

To determine the optimal ϵ, we conducted experiments with ϵ values ranging from 0.05 to 0.25 in increments of 0.05. The experimental results, presented in Figure 4, indicate that in all three domains, the

best performance was achieved when ϵ was set to 0.15, effectively balancing exploration and exploitation. Consequently, ϵ was fixed at 0.15 as the default value for all subsequent experiments.

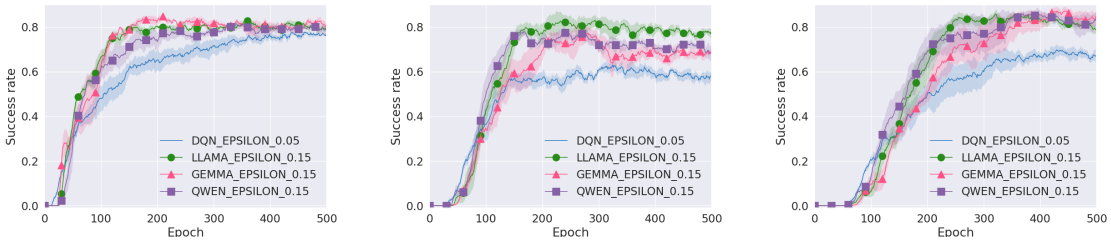
Overall, the experimental results are consistent with the findings from the ϵ hyperparameter experiments conducted for the DQN in Appendix G. Both excessively small and excessively large values of ϵ resulted in reduced exploration efficiency. However, a notable difference was observed in the optimal ϵ between the two approaches: while the best ϵ for the DQN agent was 0.05, the optimal ϵ for the LLM-driven semantic action pruning approach was 0.15. This discrepancy can be attributed to the effectiveness of semantic action pruning in LLMs, which eliminates redundant or invalid actions. Consequently, a slightly larger ϵ allows the agent to explore a broader action space while maintaining a higher proportion of high-reward actions, thereby enhancing both exploration efficiency and overall training effectiveness.



(a) Movie (b) Rest. (c) Taxi

Figure 4: Impact of LLMs Semantic Action Pruning on Exploration Rate.

E LLMs Compatibility Experiment



(a) Movie (b) Rest. (c) Taxi

Figure 5: Our approach combines the performance of different LLMs.

Agents	Taxi		Rest		Movie	
	SR↑	AS↑	SR↑	AS↑	SR↑	AS↑
DQN_EPSILON_0.0	0.4866	3.1	0.4274	2.8	0.3811	2.6
DQN_EPSILON_0.05	0.5562	3.3	0.4426	3.1	0.4860	2.8
NOISY_DQN	0.4964	2.9	0.2021	2.5	0.1876	2.4
ICM_DQN	0.4025	3.2	0.0862	1.6	0.1436	1.2
LLM_DP	0.3571	3.2	0.3684	2.9	0.2637	3.3
LLM_DP_NLG	0.2028	3.4	0.1921	3.3	0.1836	3.5
LLMSAP_LLAMA	0.6648	3.5	**0.6732**	3.4	**0.6847**	**3.7**
LLMSAP_GEMMA	**0.6836**	**3.8**	0.6391	2.9	0.5986	3.2
LLMSAP_QWEN	0.6012	3.2	0.6584	**3.5**	0.6258	3.3

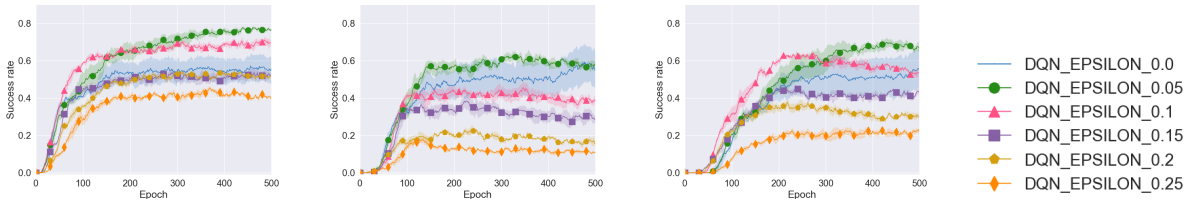
Table 2: Human evaluation results of agents in different environments.

F Human Evaluation

We conducted a blind human study with 50 students. Following the metrics of Zhao et al. (2024b) and Liu et al. (2021), we reported success rate (SR) and average score (AS, 1–5) for naturalness, coherence, and task completion. Each participant interacted with a randomly assigned domain and could terminate ineffective sessions. Retaining ≥ 20 valid dialogues per participant yielded 1,026 dialogues in total. The results (Table 2) align with the simulation experiments.

G Impact of Exploration Degree on Dialogue Policy Learning

To assess the effect of exploration on DRL-based dialogue agents and identify the optimal ϵ value, we conducted experiments utilizing the DQN algorithm, as illustrated in Figure 6. With exploration disabled (DQN_epsilon_0.0), the agent always chooses the action with the highest known reward, which restricts its ability to find the globally optimal policy, resulting in suboptimal performance. In contrast, enabling exploration enables the agent to experiment with various actions, ultimately discovering higher-reward pathways. The best results were achieved with DQN_epsilon_0.05, with performance deteriorating as ϵ increased beyond this point. This indicates that too much exploration can cause random action selection, thereby diminishing the quality of the agent's experiences. In conclusion, DRL-based dialogue agents must strike a balance between exploration and exploitation, as both insufficient and excessive exploration harm performance. Thus, DQN_epsilon_0.05 is chosen as the baseline model for our study.



(a) Movie (b) Rest. (c) Taxi

Figure 6: Effect of ϵ parameters on DQN performance.

H Prompt Design

Listing 1: Action Space.

```
[
{"diaact": "confirm_question", "inform_slots": {}, "request_slots": {}},
{"diaact": "confirm_answer", "inform_slots": {}, "request_slots": {}},
{"diaact": "thanks", "inform_slots": {}, "request_slots": {}},
{"diaact": "deny", "inform_slots": {}, "request_slots": {}},
{"diaact": "inform", "inform_slots": {"date": "PLACEHOLDER"}, "request_slots": {}},
{"diaact": "inform", "inform_slots": {"genre": "PLACEHOLDER"}, "request_slots": {}},
{"diaact": "inform", "inform_slots": {"state": "PLACEHOLDER"}, "request_slots": {}},
{"diaact": "inform", "inform_slots": {"city": "PLACEHOLDER"}, "request_slots": {}},
{"diaact": "inform", "inform_slots": {"zip": "PLACEHOLDER"}, "request_slots": {}},
{"diaact": "request", "inform_slots": {}, "request_slots": {"moviename": "UNK"}},
{"diaact": "request", "inform_slots": {}, "request_slots": {"theater": "UNK"}},
{"diaact": "request", "inform_slots": {}, "request_slots": {"starttime": "UNK"}},
{"diaact": "request", "inform_slots": {}, "request_slots":
{"diaact": "request", "inform_slots": {}, "request_slots": {"zip": "UNK"}},
{"diaact": "request", "inform_slots": {}, "request_slots": {"mpaa_rating": "UNK"}},
{"diaact": "request", "inform_slots": {}, "request_slots": {"video_format": "UNK"}},
{"diaact": "request", "inform_slots": {}, "request_slots": {"price": "UNK"}},
{"diaact": "request", "inform_slots": {}, "request_slots": {"actor": "UNK"}},
{"diaact": "request", "inform_slots": {}, "request_slots": {"description": "UNK"}},
{"diaact": "request", "inform_slots": {}, "request_slots": {"other": "UNK"}},
{"diaact": "request", "inform_slots": {}, "request_slots": {"numberofkids": "UNK"}}
...
]
```

Table 3: Descriptions of Prompts used for LLM-based baselines.

Model	Prompt
LLMSAP	You must strictly execute the following commands: 1. system roles: as an auxiliary dialogue policy module in a task-oriented dialogue system, you are required to perform semantic action pruning on the action space based on the current dialogue state, thereby assisting in policy optimization. 2. Processing user dialogue state and action space: you will receive an action space formatted similarly to Listing 1, along with a user dialogue state formatted similarly to Listing 2. You should leverage your powerful semantic understanding capabilities to deeply analyze the semantic relevance between candidate actions and the current dialogue context, to identify and eliminate invalid or redundant actions that do not match the dialogue state. 3. Generate system actions: based on the above analysis, you are expected to prune the action space and retain the actions that are more semantically relevant and potentially more rewarding. Multiple actions can be retained. The final output should be a list of indices corresponding to the retained actions in the original action space. 4. Command execution requirements: you must strictly adhere to the above instructions. The output must be a standard JSON string in the following format: `{"new_actions": [index0, index1, ...]}` All elements must be integers. Do not generate any additional text.
LLM_DP	You must strictly execute the following commands: 1. system roles: as the dialogue policy module of a task-oriented dialogue system, you need to give actions based on the current state of the dialogue. 2. Processing user dialogue state: you will receive a dialogue state in a format similar to the Listing 2 data format. This state will be used as a basis for decision-making. 3. Generate system actions: based on the user dialogue state, you need to generate system actions. These actions should be provided in the following format: [[“ActionType”, “Domain”, “Slot”, “Value”]] where ‘ActionType‘ denotes the type of action (e.g. Request, Inform, Confirm, etc.), ‘Domain‘ specifies the associated domain (e.g. restaurant, taxi, hotel, etc.), ‘Slot‘ is the specific information slot associated with the action (e.g. name, area, type, etc.), and ‘Value‘ is the corresponding value or an empty string. 4. Command execution requirements: strictly enforce the above command, the generated data must be in JSON format, and prohibit the generation of other data.
LLM_DP_NLG	You must strictly execute the following commands: 1. system roles: as the dialog policy module and natural language generation module of a task-oriented dialogue system, you need to give actions based on the current state of the dialogue. 2. Processing user dialogue state: you will receive a dialogue state in a format similar to the Listing 2 data format. This state will be used as a basis for decision-making. 3. Generate system actions: make decisions based on the state of the dialogue and generate natural language directly back to the user. 4. Command execution requirements: strictly enforce the above command, the generated data must be in JSON format, and prohibit the generation of other data.

Listing 2: Dialogue State.


```
{
"agent_action":{
"request_slots":{"moviename":"UNK"},"turn":5,"speaker":"agent","inform_slots":{},"diaact":"request"},
"user_action":{
"request_slots":{},"turn":6,"speaker":"user","inform_slots":{"moviename":"zootopia"},"diaact":"inform"},
"turn":7,
"current_slots":{
"request_slots":{"theater":"UNK"},
"agent_request_slots":{"moviename":"UNK"},
"inform_slots":{"moviename":"zootopia"},
"proposed_slots":{}},
"kb_results_dict":{"matching_all_constraints":278,"moviename":278},
"history":[
{"request_slots":{"theater":"UNK"},"turn":0,"speaker":"user","inform_slots":{"moviename":"zootopia"},"diaact":"request"},
{"request_slots":{"moviename":"UNK"},"turn":1,"speaker":"agent","inform_slots":{},"diaact":"request"},
{"request_slots":{},"turn":2,"speaker":"user","inform_slots":{"moviename":"zootopia"},"diaact":"inform"},
{"request_slots":{"moviename":"UNK"},"turn":3,"speaker":"agent","inform_slots":{},"diaact":"request"},
{"request_slots":{},"turn":4,"speaker":"user","inform_slots":{"moviename":"zootopia"},"diaact":"inform"},
{"request_slots":{"moviename":"UNK"},"turn":5,"speaker":"agent","inform_slots":{},"diaact":"request"}
  ]
}
```


I Action Space Decision-Making Algorithm

Algorithm 1: Action Space Decision-Making Algorithm.

Input: Current state s_t; pruned action space A'_t; Q-network parameters θ; exploration rate ϵ; target network update frequency τ

Output: Selected action a_t; updated parameters θ

Action Selection:
Generate a random number $r \in [0, 1]$;
if $r < \epsilon$ **then**
  Randomly select $a_t \in A'_t$;
end if
else
  $a_t \leftarrow \arg\max_{a \in A'_t} Q(s_t, a; \theta)$;
end if
Environment Interaction:
Execute action a_t; observe reward r_t and next state s_{t+1};
Q-value Update:
$y_t \leftarrow r_t + \gamma \max_{a' \in A'_t} Q(s_{t+1}, a'; \theta^-)$;
Compute loss $L(\theta) \leftarrow (y_t - Q(s_t, a_t; \theta))^2$;
Update θ by minimizing $L(\theta)$;
Target Network Update:
if *Current step* $t \mod \tau = 0$ **then**
  $\theta^- \leftarrow \theta$;
end if
return Selected action a_t, updated parameters θ;