Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Semantic Reward Shaping via Graph Abstraction and a Lightweight Language Model to Address Reward Sparsity in Task-Oriented Dialogue Systems

Shaghayegh Saffari
Semnan University
Morteza Dorrigiv
dorrigiv@semnan.ac.ir
Semnan University
Farzin Yaghmaee
Semnan University

Research Article

Keywords:

Posted Date: November 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11686673/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Semantic Reward Shaping via Graph Abstraction and a Lightweight Language Model to Address Reward Sparsity in Task-Oriented Dialogue Systems

Abstract

Reinforcement learning (RL) for task-oriented dialogue (TOD) is challenged by sparse and delayed rewards, which hinder effective policy learning and can lead to suboptimal dialogue policies. Existing dense-reward methods exploit structural and transition-based information, while effectively incorporating dialogue semantics and stage information into reward estimation remains challenging. To address these challenges, this paper proposes a potential-based reward shaping (PBRS) approach that learns cluster-level semantic potentials offline and reuses them during online RL. Dialogue states are represented by concatenating structural state features with MiniLM semantic-frame embeddings. Differentiable MinCut pooling compresses the resulting transition graph into soft behavioral clusters, while a temporal-aware graph attention encoder incorporates dialogue-stage information. An Adaptive Continuous Prompt Constructor (ACPC) conditions a LoRA-adapted MiniLM to estimate cluster potentials without handcrafted prompts. During online learning, a state's potential is computed as a soft-assignment-weighted combination of stored cluster potentials, eliminating the need for LoRA, prompt, or graph computation during policy learning. The fixed, state-dependent potentials preserve the optimal policy under the PBRS formulation. Experiments on three Microsoft Dialogue Challenge domains show that the proposed approach improves success rates over sparse-reward training by 11.6, 6.4, and 12.6 percentage points on Movie Ticket Booking, Restaurant Reservation, and Taxi Ordering, respectively. It also achieves a 1.2-point improvement over the strongest graph-based baseline on Movie Ticket Booking. Ablation studies further examine the contributions of semantic frames, temporal encoding, and ACPC. These results demonstrate that the proposed approach can effectively mitigate reward sparsity and improve policy learning in RL-based TOD systems.

Keywords: Task-oriented dialogue, Reinforcement learning, Potential-based reward shaping, Graph representation learning, Parameter-efficient language model adaptation

1 Introduction

Dialogue systems enable natural language interaction between humans and intelligent systems and are widely used in applications such as information retrieval and ticket reservation (Feng et al. 2023). These systems are generally categorized into Task-Oriented Dialogue (TOD) and Open-Domain Dialogue (ODD) systems. While ODD systems support general conversations, TOD systems focus on specific user goals, such as restaurant reservation and ticket booking, requiring accurate intent tracking and multi-turn dialogue management (Xu et al. 2019; Chen et al. 2017).

TOD systems can be implemented using pipeline-based or end-to-end architectures (Ni et al. 2023). While pipeline-based systems optimize each component independently, end-to-end approaches jointly optimize the entire dialogue process to improve dialogue policy learning (Algherairy and Ahmed 2024). Regardless of the underlying architecture, most TOD systems comprise four core components: Natural Language Understanding (NLU), Dialogue State Tracking (DST), Dialogue Policy Learning (DPL), and Natural Language Generation (NLG) (Chen et al. 2017), which typically interact with an external knowledge base during dialogue management (Ni et al. 2023).

A TOD system can be formulated as a sequential decision-making process in which an agent observes the dialogue state, selects an action, and interacts with the user until task completion. This sequential nature makes Reinforcement Learning (RL) a natural framework for dialogue policy optimization through cumulative rewards (Li et al. 2017). RL also supports flexible reward formulation and can provide more flexible policy optimization than conventional supervised learning (Ni et al. 2023). However, RL-based dialogue systems still face several challenges, particularly the problem of sparse and delayed rewards (Feng et al. 2023; Sun et al. 2025).

In RL-based dialogue systems, rewards are often determined by the overall success or failure of an episode, leaving intermediate dialogue turns with little or no feedback (Jang et al. 2022; Sun et al. 2025). Such sparse rewards make it difficult to assess the contribution of individual actions, for example, identifying relevant symptoms in medical dialogues, which may lead to inefficient or unsafe decisions (Xu et al. 2019). Combined with large state–action spaces, complex environments, and limited interaction data, sparse rewards hinder effective exploration–exploitation (Sun et al. 2025), cause cold-start problems (Durugkar et al. 2021), slow policy learning, and may prevent convergence to optimal policies (Algherairy and Ahmed 2024; Sun et al. 2025). Consequently, RL-based TOD systems often require extensive interactions to learn effective dialogue policies, limiting their practical applicability.

These challenges motivate the development of informative dense reward functions. However, manually engineered rewards, which are commonly used in RL-based dialogue systems (Li et al. 2017; Jang et al. 2022; Snell et al. 2022; Wu et al. 2019), typically rely on predefined rules and domain expertise and may fail to generalize to unobserved states and actions (Takanobu et al. 2019; Hou et al. 2021). To address this issue, implicit and dense reward mechanisms have been explored (Klissarov and Precup 2020; Zhu et al. 2023; Kamuni et al. 2024). Approaches based on human demonstrations can provide informative feedback but often require costly annotated data (Wang et al. 2020a; Ramachandran et al. 2022; Gordon-Hall et al. 2020), whereas

intrinsic-reward methods may depend on additional simulator information and introduce considerable computational complexity (Kamuni et al. 2024; Chu et al. 2024; Geishauser 2024).

Furthermore, to alleviate sparse-reward problems, reward shaping provides additional learning signals that improve exploration and policy optimization. In particular, Potential-Based Reward Shaping (PBRS) can accelerate learning and convergence while preserving the optimal policy (Sang et al. 2023a,b; Gerasyov and Makarov 2023; Su et al. 2015). In RL-based dialogue systems, PBRS provides informative signals at intermediate state transitions, facilitating more efficient policy learning (Su et al. 2015). However, the large action spaces of dialogue systems can still hinder convergence, motivating further improvements (El Asri et al. 2013). Early approaches mainly relied on manually designed heuristics or rewards inferred during training (Zhu et al. 2023; Kamuni et al. 2024; Zhao et al. 2020), while recent studies have explored graph-based reward propagation, representing dialogue states as nodes and transitions as edges to learn informative behavioral representations for dense reward estimation (Saffari et al. 2025).

Graph representation learning is well suited to this setting because it captures structural dependencies in complex systems and naturally reflects the state-transition structure of Markov decision processes (MDPs) (Gerasyov and Makarov 2023). It has also demonstrated effectiveness in applications such as activity recommendation (Gong and Fang 2026), fault diagnosis (Wang et al. 2026b), and decision-making (Saffari et al. 2025). Recent studies have further demonstrated the effectiveness of graph-based reinforcement learning in complex decision-making problems. For example, Lu et al. developed a graph-based RL approach for software-defined networking traffic engineering, using graph neural networks to represent network topology and support traffic allocation decisions (Lu et al. 2025). Similarly, Wang et al. integrated graph multi-head attention into a deep RL architecture to capture complex network-state dependencies and optimize routing decisions through a multi-objective reward function (Wang et al. 2026a). However, the potential of graph-based RL for addressing sparse rewards in RL-based TOD systems remains largely underexplored.

Despite these advances, two important limitations remain. First, existing graph-based reward shaping methods mainly capture structural relationships among dialogue states while overlooking their temporal evolution (Saffari et al. 2025). Consequently, states with similar local graph structures but different temporal contexts may receive similar representations despite having different implications for eventual task success, limiting the discriminative capability of reward modeling, particularly in longer dialogues. Second, although pretrained language models can provide compact and informative semantic representations (Peyton and Unnikrishnan 2023), existing reward-learning approaches (Feng et al. 2023) have not jointly integrated semantic information with graph-based structural and temporal representations for reward learning in TOD. While recent studies have demonstrated the complementary benefits of combining graph representations with semantic modeling in other applications (Zhang and Liu 2026), such integration remains largely unexplored for reward learning in RL-based TOD systems.

Recent studies have also explored the use of semantic and structured knowledge to improve reinforcement learning in interactive recommendation systems. Fan et al. (Fan et al. 2026) proposed an adaptive RL approach that employs large language models for offline policy pretraining and knowledge graphs to guide online candidate generation, thereby addressing sample-efficiency challenges in interactive recommendation. Similarly, Tiwary et al. (Tiwary et al. 2025) investigated the integration of knowledge graph embeddings and reinforcement learning to support explainable recommendation and preference-aware decision-making. These studies demonstrate the relevance of incorporating semantic and structured knowledge into RL-based decision-making; however, they do not address the problem of learning temporally informed, state-dependent reward potentials for sparse-reward TOD systems.

Moreover, relying on large generative language models for reward estimation can introduce substantial computational overhead (Feng et al. 2023; Xu et al. 2025; Snell et al. 2022). Therefore, an efficient reward-learning framework that jointly exploits structural, temporal, and semantic information while remaining computationally lightweight is still needed.

To address these limitations, this paper proposes Semantic, Structural, and Temporal-aware Reward Shaping (SST-RS), a PBRS approach that jointly exploits structural, semantic, and temporal characteristics of dialogue behavior.

Relative to Saffari et al. (2025), in which every node is described solely by a 224-dimensional state vector (which slots are filled, which actions occurred, and the turn information), SST-RS additionally extracts the meaning of each state from its semantic frame (intent, dialogue acts, slot values) with a pretrained encoder, so that states that are structurally identical but differ in meaning can be separated.

SST-RS first constructs a semantic dialogue transition graph by combining structural dialogue-state representations with semantic frame embeddings, allowing both transition relationships and semantic similarities to be captured. Differentiable Min-Cut pooling (Bianchi et al. 2020, 2019) then performs soft, feature-driven clustering to obtain compact behavioral representations. This abstraction reduces redundancy and computational complexity (Bianchi et al. 2020, 2019) and establishes a shared behavioral space through which reward information can be generalized across related states. Because the assignment is computed from state features rather than graph connectivity alone, newly observed states can in principle be associated with learned behavioral clusters even when they have few or no graph neighbors (Saffari et al. 2025); we discuss this property as a design motivation and examine it in Section 4.2.1.

However, graph abstraction alone does not explicitly characterize how dialogue behavior evolves across successive turns. To address this limitation, a Temporal Graph Attention Encoder (TGAE) enriches the behavioral clusters with temporal dependencies through sinusoidal positional encoding and attention-based message passing. This allows the model to distinguish behavioral regions that exhibit similar structural and semantic characteristics but occur at different stages of a dialogue, providing a more discriminative basis for reward estimation, particularly in longer dialogues.

The resulting representations are transformed by an Adaptive Continuous Prompt Constructor (ACPC) into continuous prompt embeddings (Lester et al. 2021; Liu et al. 2022; Li and Liang 2021) that condition a LoRA-adapted (Hu et al. 2022) MiniLM

encoder (Wang et al. 2020b). This enables efficient offline semantic potential learning without handcrafted prompts or computationally expensive generative language models. During online RL, each newly observed state is softly assigned to the learned behavioral clusters, and its potential is obtained as a weighted combination of cluster-level potentials. The resulting potential is incorporated into PBRS to provide dense intermediate rewards while preserving policy invariance.

To validate the effectiveness of the proposed approach, experiments are conducted on three benchmark TOD domains from the Microsoft Dialogue Challenge, namely Movie Ticket Booking, Restaurant Reservation, and Taxi Ordering. Results show consistent improvements in dialogue success rate, cumulative reward, and dialogue efficiency over sparse-reward training and several state-of-the-art dense-reward baselines.

The main contributions of this paper are as follows:

- An SST-RS approach is proposed to address reward sparsity in RL-based TOD by incorporating structural, semantic, and temporal dialogue information into potential-based reward shaping.
- A MinCut-based behavioral graph abstraction with feature-driven soft assignment is used to learn compact behavioral clusters and share cluster-level reward potentials across related dialogue states.
- A Temporal Graph Attention Encoder (TGAE) is developed to capture dialogue-stage information and distinguish structurally and semantically similar behavioral regions occurring in different contexts.
- An Adaptive Continuous Prompt Constructor (ACPC) is proposed to bridge graph-derived behavioral representations with a LoRA-adapted MiniLM encoder, enabling parameter-efficient semantic potential learning without handcrafted prompts or large language models (LLMs).
- A policy-invariance result is given for the fixed cluster potentials: because the potential is only retrieved and never updated during policy learning, the optimal policy of the underlying MDP is preserved (Section 3.2.8).
- Experiments on three benchmark TOD domains show improvements in task success, cumulative reward, and dialogue efficiency over state-of-the-art dense reward and policy-learning baselines, with ablation studies examining the main components.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the proposed methodology. Section 4 reports the experimental setup, results, and discussion. Section 5 concludes the paper and discusses future work.

2 Related Works

2.1 Addressing reward sparsity in dialogue systems

Reward sparsity remains a fundamental challenge in RL for dialogue systems, as informative feedback is often concentrated at the end of a dialogue. Existing studies have therefore explored different mechanisms for providing additional feedback to intermediate states. These approaches can be broadly viewed according to whether the

additional reward is derived from external supervision and interaction data or from intrinsic properties of the dialogue trajectories.

A major line of research derives additional rewards from human demonstrations, preferences, or real user interactions. Wang et al. (2020a) used expert demonstrations to guide policy learning through imitation-based reward signals, while Gordon-Hall et al. (2020) incorporated expert dialogue acts into the replay process. S2Agent combines policy and reward shaping using imitation models trained from human demonstrations (Wang et al. 2020a). CASPI instead uses causal and preference-based information to shape rewards toward human goals (Ramachandran et al. 2022). These approaches can provide informative guidance during early policy learning, but their effectiveness depends on the availability and quality of demonstrations, preferences, or real interaction data, which increases acquisition cost and may introduce bias or limit cross-domain generalization.

Other approaches generate additional rewards directly from dialogue trajectories or simulated experience, reducing dependence on explicit human supervision. Dynamic reward learning with Dueling Dyna-Q estimates rewards from real and simulated interactions (Zhao et al. 2020), but requires access to user data and additional environment modeling. Hindsight-based methods such as LHUA (Lu et al. 2021) exploit unsuccessful interactions by relabeling dialogue goals. Intrinsic-reward approaches further encourage exploration using state novelty, curiosity, or prediction errors (Kamuni et al. 2024). Similarly, FeudalGain uses information gain as an auxiliary reward (Geishauser 2024), while MACRM combines task rewards with curiosity derived from prediction errors (Sun et al. 2025). Although these methods improve exploration and sample efficiency, their auxiliary signals may require additional objectives and hyperparameters and do not necessarily quantify the long-term contribution of an individual dialogue state to task success.

Preference- and trajectory-based reward learning provides another way to densify feedback without manually specifying rewards. FanReward learns reward models from trajectory-level scores and their relative rankings (Feng et al. 2023), with ranking-based learning showing improved robustness to noisy trajectory evaluations. However, such approaches may become computationally expensive as the state–action space grows. More generally, inverse and adversarial reward-learning methods can infer dense rewards from observed dialogue behavior (Takanobu et al. 2019; Hou et al. 2021), but their reliance on discriminators, human data, or repeated retraining can limit scalability in dynamic environments.

Reward shaping methods provide a complementary solution by augmenting the original environmental reward rather than replacing it. Earlier studies employed manually designed or learned potential functions to distribute sparse feedback across dialogue turns (Su et al. 2015), while other approaches incorporated auxiliary information, such as inferred human evaluation scores and user social signals, into the shaping process (El Asri et al. 2013; Ferreira and Lefevre 2013). More recent preference- and causally informed shaping methods, such as CASPI (Ramachandran et al. 2022), further attempt to align the additional reward with human-defined task objectives. These methods can accelerate policy optimization while retaining the original task objective; however, the quality of the resulting guidance depends strongly on how the potential

or auxiliary reward is constructed and, in some cases, on external supervision. State abstraction and bisimulation (Li et al. 2006) group states that are equivalent with respect to reward and transition behavior, which motivates cluster-level potentials; SST-RS instead uses a soft, feature-driven assignment learned from graph structure and semantics rather than exact equivalence. Potentials defined through value functions are known to give effective shaping (Ng et al. 1999), and reward-redistribution methods such as RUDDER (Arjona-Medina et al. 2019) reassign delayed returns to earlier steps; in contrast, SST-RS learns a fixed cluster-level potential offline from semantic-frame and graph information.

Overall, existing approaches alleviate reward sparsity through external supervision, added interaction or simulation, intrinsic signals, or hand-crafted and learned shaping, typically at the cost of expensive supervision or added computational overhead. Moreover, most assign reward at the level of individual states or local state–action pairs, with limited capacity to capture how structurally related states differ semantically and temporally. This motivates reward modeling that jointly exploits structural, semantic, and temporal characteristics of dialogue states without heavy reliance on human supervision or online interaction.

2.2 Graph representation learning and language models for reward modeling

Graph representation learning provides a natural mechanism for modeling relationships among dialogue states and propagating reward information across connected states. Early studies investigated graph-based reward propagation in game environments. For example, Klissarov and Precup (2020) explored graph-based reward propagation for sparse-reward navigation tasks, while Gerasyov and Makarov (2023) modeled visited states as a graph and diffused reward information through state relationships. Directed graph structures were subsequently investigated to better represent asymmetric transitions and reward propagation (Sang et al. 2023a). Hierarchical graph formulations have also been proposed to propagate reward information across different levels of state representation (Sang et al. 2023b). Although these studies demonstrate the potential of graph structures for alleviating sparse rewards, they primarily focus on game or navigation environments rather than the semantic and sequential characteristics of TOD systems.

In the context of dialogue systems, Saffari et al. (2025) introduced a graph representation-based PBRS approach specifically for TOD. The method represents each dialogue state solely by its 224-dimensional structural state vector and performs soft clustering based on structural similarity and transition relationships. Reward information is then propagated through the resulting behavioral representation using graph-based models such as GraphSAGE, Graph Attention Networks (GAT), and Node2Vec. This approach improves reward propagation and policy learning compared with conventional sparse-reward training. However, its state representation is primarily driven by structural and graph information and does not explicitly model the semantic content of dialogue states. Moreover, the temporal context in which a state occurs is not explicitly modeled, limiting the ability to distinguish behaviorally similar states that contribute differently to dialogue success at different stages of an interaction.

Language-model-based methods provide a complementary perspective by representing dialogue states according to their semantic content. Siddique et al. (2022) used sentence embeddings to derive reward signals from semantic similarity. More recent approaches employ language models to generate or evaluate dense reward signals. Xie et al. (2024) uses an LLM to generate executable reward functions, while Chan et al. (2024) redistributes sequence-level rewards into finer-grained signals using attention-based attribution. Similarly, Quadros et al. (2025) uses language-model-based state evaluation to construct an intrinsic reward. Although language-model-based methods can capture dialogue semantics, repeated LLM inference for reward estimation can incur substantial computational costs, limiting their scalability in RL.

Parameter-efficient adaptation offers a more practical alternative. Prompt-based approaches such as prompt tuning (Lester et al. 2021), P-tuning (Liu et al. 2022), and prefix-tuning (Li and Liang 2021) adapt pretrained models without updating all model parameters. LoRA further reduces the number of trainable parameters by introducing low-rank adaptation matrices (Hu et al. 2022). Compact pretrained encoders such as MiniLM (Wang et al. 2020b) therefore provide an attractive basis for semantic representation learning without the cost of full-scale LLM inference.

Overall, graph-based methods provide efficient structural modeling and reward propagation, but offer limited semantic grounding and temporal discrimination. Language-model-based methods capture richer dialogue semantics, but generally do not explicitly represent the relational and temporal organization of dialogue states and may impose substantial computational costs. These complementary limitations motivate a unified reward-learning framework that combines structural, semantic, and temporal information while remaining computationally efficient.

3 Proposed method

This section first introduces the overall architecture of the end-to-end deep RL-based TOD system used in this work. It then presents the proposed semantic reward shaping approach in a stage-wise manner.

3.1 RL-based dialogue system

In this work, the dialogue management problem is formulated as an MDP (Li et al. 2017), $M = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, where $\mathcal{S}$ denotes the dialogue state space, $\mathcal{A}$ the action space, $\mathcal{R}$ the reward function, $\mathcal{P}$ the transition probability, and γ the discount factor. The objective of the dialogue policy learning (DPL) module is to learn an optimal policy π^* that maximizes the expected cumulative discounted reward, as in Eq. (1).

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi}\Big[\sum_{t=0}^{\infty} \gamma^t r_t\Big] \tag{1}$$

where r_t denotes the reward obtained at dialogue turn t. Since this work focuses on reward shaping for dialogue policy optimization rather than natural language understanding and generation, we adopt the simplified end-to-end dialogue environment introduced by Li et al. (2017). In this environment, the NLU and NLG components

are omitted, and the interaction is performed through structured semantic frames. To simulate realistic understanding errors, an error model controller introduces slot-level noise during training, including random noise, slot deletion, incorrect slot values, and incorrect slot assignments (Li et al. 2017).

The dialogue agent interacts with a rule-based user simulator through a sequence of semantic frames. Each semantic frame consists of an intent describing the dialogue action type, inform slots representing user constraints or provided information, and request slots indicating the information required from the system. An example of the semantic frame representation for modeling the user goal is illustrated in Fig. 1. Based on the dialogue history, previous user/system actions, database information, and the current semantic frame, the dialogue state tracker (DST) constructs a compact state representation, which is then provided to the DPL module for action selection. The dialogue state encodes the contextual information required for decision making, including the semantic frames of user and system actions, inform and request slots, current slot status, database information, and dialogue progress. These features are represented as a fixed-length multi-hot vector. The turn representation provides information about the dialogue progress and the proximity to the maximum allowed dialogue length. The complete state feature representation is illustrated in Fig. 2. The DPL module is implemented using a Double Dueling Deep Q-Network (D3QN), following Saffari et al. (2025). Due to the large state-action space, D3QN approximates the action-value function using deep neural networks instead of tabular learning. The Double Q-learning mechanism reduces Q-value overestimation, while the dueling architecture separates the estimation of state value and action advantage, improving learning stability. Experience replay, a target network, and an ϵ-greedy exploration strategy are also employed to stabilize policy optimization.


```
{
 "request_slots": {"date": "UNK", "theater": "AMC", "starttime": "UNK" },
 "intent": "request",
 "inform_slots": {"numberofpeople": "3",  "moviename": "the_batman" }
}
```


Fig. 1 Example of a user semantic frame containing the dialogue intent, provided information slots, and requested information slots.

$$s_t = \begin{bmatrix} UserAction, UserInformSlots, UserRequestSlots, AgentAction, \\ AgentInformSlots, AgentRequestSlots, CurrentSlotStatus, DialogueTurn, \\ One-hotTurn, BinaryDatabaseFeatures, DatabaseEntityCount \end{bmatrix}$$

Fig. 2 Feature representation of the dialogue state used by the RL agent.

During the initial training phase, the policy is optimized using the sparse environmental reward function adopted from Li et al. (2017), as in Eq. (2).

$$r_t = \begin{cases} -1 + 2L_{\max} & \text{if success} \\ -1 - L_{\max} & \text{if failure} \\ -1 & \text{otherwise} \end{cases} \tag{2}$$

where $L_{\max}$ is the maximum number of turns (40 for Movie Ticket Booking and 30 for the other domains), successful and failed dialogues receive terminal rewards, and each intermediate turn receives a constant penalty. Since informative feedback is mainly available at the end of dialogues, intermediate states suffer from limited supervision, leading to the sparse reward problem. To address this limitation, SST-RS learns semantic potentials from offline dialogue trajectories and incorporates them into the reward function to provide dense guidance during RL.

3.2 Addressing reward sparsity using SST-RS approach

As discussed earlier, in TOD systems with sparse rewards, intermediate dialogue states receive no direct feedback, making it difficult for the policy to identify promising behaviors. The proposed SST-RS approach learns semantic potentials from dialogue state transitions and incorporates them into a PBRS framework to provide informative rewards for intermediate states while preserving the optimal policy. The learned reward knowledge is transferred to newly observed dialogue states through their behavioral representations and used during online policy optimization. The main stages are described in the following subsections.

3.2.1 Construction of the semantic dialogue transition graph

The dialogue state space in TOD systems is extremely large and sparse (Feng et al. 2023), making it impractical to collect and construct the complete state-transition graph. Therefore, the proposed framework relies on a sufficiently large sample of observed dialogue transitions to construct a semantic dialogue transition graph, which is subsequently compressed into a compact behavioral graph for efficient reward learning.

To obtain representative transition samples, the dialogue agent is first trained with the original sparse rewards until the success rate exceeds a predefined threshold (Section 4.1.2). Subsequent dialogue transitions are then collected, including the state vector, semantic frame, and corresponding environmental reward, thereby reflecting the natural behavioral distribution of the trained policy. All transitions are collected once the threshold is reached, regardless of the outcome of individual episodes.

Each dialogue episode is represented as a temporal sequence, $\mathcal{T} = \{s_1, s_2, \ldots, s_T\}$, where s_t denotes the dialogue state at interaction step t. The dialogue transition graph $G = (V, E)$ is constructed from the collected trajectories, where dialogue states are represented as 224-dimensional vectors $s_i \in \mathbb{R}^{d_s}$ (binary slot features plus a few non-binary attributes such as turn information). A state containing exactly three active slots is added to the node set V and serves as the starting point of a transition chain.

The subsequent dialogue states are then connected sequentially by adding an edge (s_t, s_{t+1}) for each observed transition. When another state with exactly three active slots is encountered, the current transition chain is terminated, and no edge is added between this state and the preceding state; the subsequent transitions are treated as a new chain. This construction preserves the observed sequential transition structure while avoiding connections across stable goal states. Isolated nodes are retained to preserve rare or irregular dialogue behaviors. Edges are undirected and link states that appeared consecutively within an episode. The three-active-slot rule relies on the structure of these domains: every episode starts with one inform slot and one request slot, and the inform slot is also reflected in the current-slot features, which yields exactly three active slot features in the initial state. The rule is therefore a structural marker of episode starts that is specific to these domains; other domains would require an analogous marker.

Although the structural state vector captures information such as slot completion, it does not adequately represent semantic relationships. Therefore, the framework incorporates the semantic frame associated with each dialogue state, including user intents, dialogue acts, domains, and slot-value information. Each semantic frame is first serialized into a textual representation listing the intent, the inform slots with their values, and the request slots (e.g., "intent: request; inform: moviename = ...; request: starttime") and then tokenized using the pretrained tokenizer of MiniLM-L12-H384-uncased (Wang et al. 2020b). The token sequence is fed into the pretrained MiniLM encoder to generate contextual token embeddings, and a fixed-length semantic representation is obtained by mean pooling (Reimers and Gurevych 2019), as in Eq. (3).

$$e_i^{sem} = \text{MeanPool}\left(f_{\text{MiniLM}}\left(\text{Tokenizer}(SF_i)\right)\right) \tag{3}$$

where SF_i denotes the serialized semantic frame of the i-th dialogue state, $\text{Tokenizer}(\cdot)$ is the pretrained MiniLM tokenizer, $f_{\text{MiniLM}}(\cdot)$ denotes the pretrained MiniLM-L12-H384-uncased encoder, $\text{MeanPool}(\cdot)$ represents mean pooling over contextual token embeddings, and $e_i^{sem} \in \mathbb{R}^{384}$ is the resulting semantic embedding.

The structural and semantic representations are fused to construct the final node representation, $x_i = [s_i \parallel e_i^{sem}]$, where $s_i \in \mathbb{R}^{d_s}$ ($d_s = 224$) denotes the structural state representation, $e_i^{sem} \in \mathbb{R}^{d_m}$ ($d_m = 384$) is the semantic embedding, and $\parallel$ denotes concatenation, so that $x_i \in \mathbb{R}^d$ with $d = d_s + d_m = 608$. The non-binary attributes of s_i are scaled to $[0, 1]$ so that all structural features share a comparable range. Finally, the node feature matrix $X = [x_1, x_2, \ldots, x_N]^\top$ and the adjacency matrix $A \in \mathbb{R}^{N \times N}$ are constructed from the dialogue transition graph. To improve computational efficiency, the adjacency matrix is stored in sparse format and processed using PyTorch Geometric. The resulting graph jointly preserves the observed transition topology and the contextual semantic representations of dialogue states.

3.2.2 Semantic state abstraction via differentiable MinCut pooling

Although the constructed graph captures observed structural transitions and contextual semantics, the underlying dialogue state space remains large and sparse, making

direct representation learning over the sampled graph computationally demanding. Moreover, different dialogue states may exhibit similar structural, semantic, or behavioral patterns and can therefore benefit from shared reward knowledge.

To address these challenges, the framework employs a differentiable MinCut pooling layer (Bianchi et al. 2020) to compress the sampled state graph into a compact behavioral graph through soft clustering. MinCut pooling uses a learnable feature-driven assignment, allowing each state to be softly associated with behavioral clusters based on its representation rather than relying solely on graph connectivity. The input is the node feature matrix $X \in \mathbb{R}^{N \times d}$ whose rows are the fused node representations x_i. A soft cluster assignment matrix is computed as in Eq. (4) (Bianchi et al. 2019).

$$S = \mathrm{Softmax}\left(XW_s\right) \tag{4}$$

where $W_s \in \mathbb{R}^{d \times K}$ is the assignment weight matrix and $S \in \mathbb{R}^{N \times K}$ represents the probability assignment of N dialogue states to K latent behavioral clusters. Each row of S is a probability distribution over clusters, so each state may contribute to several clusters. The latent representation of each behavioral cluster is obtained by weighted aggregation of the state features, as in Eq. (5).

$$Z = \mathrm{diag}\big(S^\top \mathbf{1}_N + \varepsilon\big)^{-1} S^\top X \tag{5}$$

where $Z \in \mathbb{R}^{K \times d}$ contains the cluster representations, which are optimized jointly with the downstream reward prediction objective. Unlike the weighted sum of the original MinCutPool, Z is the assignment-weighted mean of the state features, so the scale of z_i does not depend on the soft mass $\sum_n S_{ni}$ of cluster i. A small constant $\varepsilon = 10^{-8}$ is added to the denominators of Eqs. (5), (9), and (29) to handle near-empty clusters. The graph is simultaneously coarsened as in Eq. (6).

$$A_c = S^\top A S, \qquad (A_c)_{ii} \leftarrow 0 \tag{6}$$

where A_c is the adjacency matrix of the behavioral graph; as in the original MinCutPool, its diagonal is set to zero. The clustering is guided by the differentiable MinCut objective (Bianchi et al. 2019), as in Eq. (7).

$$\mathcal{L}_{\mathrm{MinCut}} = -\frac{\mathrm{Tr}\left(S^\top A S\right)}{\mathrm{Tr}\left(S^\top D S\right)} \tag{7}$$

where D is the degree matrix of the original graph. Minimizing this objective encourages strongly connected states to be assigned to the same cluster. To prevent degenerate solutions and maintain balanced assignments, an orthogonality regularization term is applied (Bianchi et al. 2019), as in Eq. (8).

$$\mathcal{L}_{\mathrm{orth}} = \left\| \frac{S^\top S}{\|S^\top S\|_F} - \frac{I_K}{\sqrt{K}} \right\|_F \tag{8}$$

where I_K is the $K \times K$ identity matrix and $\|\cdot\|_F$ is the Frobenius norm. After abstraction, the transition graph is compressed into a smaller behavioral graph that preserves essential semantic and structural patterns while reducing noise and computational complexity (Bianchi et al. 2019). The resulting shared cluster space is subsequently processed by the temporal graph attention encoder.

3.2.3 Temporal Graph Attention Encoder (TGAE)

Following graph abstraction, each node represents a behavioral cluster containing dialogue states with similar structural and semantic characteristics. Although the temporal order of states is preserved during graph construction, the cluster representations do not explicitly encode the temporal position of these behavioral regions. The proposed TGAE integrates temporal information with attention-based message passing over the behavioral graph, producing context-aware representations for reward estimation. The TGAE consists of two graph attention layers and incorporates positional encoding to represent the temporal positions of behavioral clusters.

Let $Z = \{z_1, z_2, \ldots, z_K\}$ denote the output of the MinCut pooling module, where $z_i \in \mathbb{R}^d$ is the embedding of the i-th behavioral cluster. Each state is associated with a normalized temporal position $\tau_n = t_n / L_{\max} \in [0, 1]$, and the temporal position of each cluster is obtained through the soft assignment matrix as a weighted average, Eq. (9).

$$\tau_i = \frac{\sum_{n=1}^{N} S_{ni} \tau_n}{\sum_{n=1}^{N} S_{ni} + \varepsilon} \tag{9}$$

Because τ_i is a soft average, a cluster whose states span several dialogue stages receives an intermediate position. The TGAE is therefore described as stage-aware rather than as a model of full transition sequences. Before temporal fusion, the cluster representation is mapped to the hidden size d_h of the encoder by the linear projection of the pooling module,

$$\tilde{z}_i = W_p z_i + b_p, \qquad W_p \in \mathbb{R}^{d_h \times d},\ d_h = 128. \tag{10}$$

Since a scalar τ_i cannot be directly combined with the d_h-dimensional cluster embeddings, it is mapped into a multi-frequency d_h-dimensional encoding via a sinusoidal transformation (Vaswani et al. 2017), Eq. (11). To obtain a step-scale position, τ_i is rescaled by $L_{\max}$.

$$\tilde{\tau}_i = L_{\max}\, \tau_i, \quad p_i^{(2k)} = \sin\Big(\frac{\tilde{\tau}_i}{10000^{2k/d_h}}\Big), \quad p_i^{(2k+1)} = \cos\Big(\frac{\tilde{\tau}_i}{10000^{2k/d_h}}\Big) \tag{11}$$

where d_h denotes the embedding dimension of the encoder and $k = 0, \ldots, d_h/2 - 1$ indexes the embedding components. An adaptive gate controls the contribution of semantic and temporal information, Eqs. (12) and (13).

$$\boldsymbol{\eta}_i = \sigma\left(W_t\left[\tilde{z}_i \parallel p_i\right] + b_t\right) \tag{12}$$

$$h_i^{(0)} = \boldsymbol{\eta}_i \odot \tilde{z}_i + (1 - \boldsymbol{\eta}_i) \odot p_i \tag{13}$$

where $W_t \in \mathbb{R}^{d_h \times 2d_h}$ and $b_t \in \mathbb{R}^{d_h}$ are trainable parameters, $\sigma(\cdot)$ is the sigmoid function, and $\odot$ is element-wise multiplication. Message propagation is performed over the behavioral graph using the normalized adjacency matrix (Kipf and Welling 2016), Eqs. (14) and (15).

$$\widetilde{A} = D_c^{-\frac{1}{2}} \left(A_c + I\right) D_c^{-\frac{1}{2}} \tag{14}$$

$$D_{c,ii} = \sum_j \left(A_c + I\right)_{ij} \tag{15}$$

where I is the identity matrix and $D_{c,ii}$ is the degree of the i-th cluster. Since neighboring clusters contribute unequally, a learnable attention mechanism estimates their relative importance (Velickovic et al. 2017). To avoid normalizing the adjacency twice, the normalized connectivity enters the attention as an additive bias, Eqs. (16) and (17), for layers $l = 0, 1$.

$$\xi_{ij}^{(l)} = \mathrm{LeakyReLU}\Big(a^\top \left[W^{(l)} h_i^{(l)} \parallel W^{(l)} h_j^{(l)}\right]\Big) \tag{16}$$

$$\alpha_{ij}^{(l)} = \frac{\exp\big(\xi_{ij}^{(l)} + \log \widetilde{A}_{ij}\big)}{\sum_{k \in \mathcal{N}(i)} \exp\big(\xi_{ik}^{(l)} + \log \widetilde{A}_{ik}\big)} \tag{17}$$

where $W^{(l)}$ is a learnable linear transformation, a is a trainable attention vector, and $\mathcal{N}(i) = \{j : \widetilde{A}_{ij} > 0\} \cup \{i\}$ is the neighborhood of cluster i (since A_c is dense after soft pooling, the neighborhood in practice covers all other clusters, as in the fully connected pooled graph with $K = 30$). The aggregated message and the adaptive propagation gate are given by Eqs. (18) and (19).

$$m_i^{(l)} = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} W^{(l)} h_j^{(l)} \tag{18}$$

$$g_i^{(l)} = \sigma\left(W_g^{(l)} \left[h_i^{(l)} \parallel m_i^{(l)}\right] + b_g^{(l)}\right) \tag{19}$$

where $W_g^{(l)} \in \mathbb{R}^{d_h \times 2d_h}$ and $b_g^{(l)} \in \mathbb{R}^{d_h}$ are trainable parameters, $W^{(l)} \in \mathbb{R}^{d_h \times d_h}$, and $a \in \mathbb{R}^{2d_h}$. The representation is updated with a residual connection and layer normalization (Vaswani et al. 2017), Eq. (20), and the final representations are collected as H, Eq. (21).

$$h_i^{(l+1)} = \mathrm{LayerNorm}\left(h_i^{(l)} + g_i^{(l)} \odot \mathrm{Drop}\left(\mathrm{GELU}(m_i^{(l)})\right)\right), \quad l = 0, 1 \tag{20}$$

$$H = \left\{h_1^{(2)}, h_2^{(2)}, \ldots, h_K^{(2)}\right\} \tag{21}$$

where $\mathrm{Drop}(\cdot)$ denotes dropout. H jointly encodes semantic characteristics, graph structural dependencies, and temporal information. The encoder does not directly estimate reward values; it learns behavior-aware latent representations that become reward-aware through end-to-end optimization. In the remainder, $h_i \in \mathbb{R}^{d_h}$ denotes $h_i^{(2)}$ and $H \in \mathbb{R}^{K \times d_h}$.

3.2.4 ACPC

The TGAE produces contextual behavioral representations by integrating structural, semantic, and temporal information across the behavioral graph. However, these representations primarily encode the behavioral context of each cluster and do not explicitly preserve the detailed semantic content of the underlying states. To address this limitation, ACPC converts each behavioral representation into continuous prompts that condition a LoRA-adapted MiniLM encoder together with a representative semantic frame. The semantic frame provides the explicit dialogue content of the cluster, while the TGAE representation provides its structural, behavioral, and temporal context. MiniLM-L12-H384 is adopted as a lightweight Transformer backbone (Wang et al. 2020b), while LoRA enables parameter-efficient adaptation (Hu et al. 2022).

Let $h_i \in \mathbb{R}^{d_h}$ denote the final contextual representation of the i-th cluster. Given the fused state representation $x_j = [s_j \parallel e_j^{sem}]$ and the pre-temporal cluster representation z_i, the most representative state is selected using cosine similarity, Eq. (22).

$$j_i^* = \arg\max_j \frac{1}{2}\left[\cos\left(s_j, z_i^{(s)}\right) + \cos\left(e_j^{sem}, z_i^{(e)}\right)\right] \tag{22}$$

where $z_i = [z_i^{(s)} \parallel z_i^{(e)}] \in \mathbb{R}^d$ splits the pre-temporal cluster representation into its structural part ($d_s = 224$) and semantic part ($d_m = 384$), so that x_j and z_i both lie in $\mathbb{R}^{608}$, and the semantic frame of the selected state is denoted SF_i^{rep}. Because the two blocks may differ in scale, the cosine similarity is computed per block, which is equivalent to L2-normalizing each block before comparing x_j and z_i and gives both blocks equal weight. The ACPC projects the TGAE representation into a prompt-specific latent space and then generates L continuous prompt vectors, Eqs. (23)–(25).

$$c_i = \mathrm{GELU}(W_c h_i + b_c) \tag{23}$$

$$\Pi_i = f_{\text{prompt}}(c_i) \in \mathbb{R}^{L \times d_e} \tag{24}$$

$$\Pi_i = \{v_i^{(1)}, v_i^{(2)}, \ldots, v_i^{(L)}\} \tag{25}$$

where $W_c \in \mathbb{R}^{d_c \times d_h}$ and $b_c \in \mathbb{R}^{d_c}$ are trainable with $d_c = 128$, f_{prompt} is a linear layer from $\mathbb{R}^{d_c}$ to $\mathbb{R}^{Ld_e}$ whose output is reshaped to $L \times d_e$, $L = 12$ is the prompt length, $d_e = d_m = 384$ is the MiniLM embedding dimension, and $v_i^{(l)} \in \mathbb{R}^{d_e}$ is a continuous prompt vector. Unlike manually designed textual prompts, these prompts are generated directly from the learned behavioral representations. They are concatenated with the embedding sequence of the representative frame, and the LoRA-adapted MiniLM produces the contextual representation, Eqs. (26) and (27).

$$I_i = [\Pi_i; \mathrm{Emb}(SF_i^{rep})] \tag{26}$$

$$u_i = f_{\text{MiniLM}}(I_i; \theta_{\text{LoRA}}) \tag{27}$$

where $\mathrm{Emb}(SF_i^{rep}) \in \mathbb{R}^{T_i \times d_e}$ is the token-embedding sequence of the T_i frame tokens, so that $I_i \in \mathbb{R}^{(L+T_i) \times d_e}$, $u_i \in \mathbb{R}^{d_u}$ ($d_u = d_m = 384$) is the pooled contextual representation (obtained by mean pooling over the output tokens of the semantic-frame

positions; the prompt positions are excluded), and θ_{LoRA} denotes the LoRA parameters. The semantic reward of each cluster is estimated by a lightweight regression head, Eq. (28).

$$\widehat{\rho}_i = W_2\ \text{Drop}\big(\ \text{ReLU}(W_1 u_i + b_1)\big) + b_2 \tag{28}$$

where $W_1 \in \mathbb{R}^{128\times d_u}$, $b_1 \in \mathbb{R}^{128}$, $W_2 \in \mathbb{R}^{1\times 128}$, and $b_2 \in \mathbb{R}$ are trainable (collected in θ_{Head}). The target reward of each cluster is the soft aggregation of the environmental rewards of its assigned states, Eq. (29), where $\text{sg}(\cdot)$ denotes stop-gradient so that the assignment cannot reduce the loss by collapsing the targets.

$$\rho_i = \frac{\sum_j \text{sg}(S_{ji})\, r_j^{env}}{\sum_j \text{sg}(S_{ji}) + \varepsilon} \tag{29}$$

Policy invariance (Section 3.2.8) holds for any fixed state-dependent potential, so the choice of target affects the speed and stability of learning rather than the optimal policy. A value-like alternative, the discounted return-to-go $y_j = \sum_{k\geq 0} \gamma^k r_{j+k}^{env}$ within the episode, is compared with the target above in Table 4. The reward prediction loss is given in Eq. (30).

$$\mathcal{L}_{reward} = \frac{1}{K}\sum_{i=1}^{K} (\widehat{\rho}_i - \rho_i)^2 \tag{30}$$

Minimizing this objective jointly optimizes the ACPC, LoRA parameters, regression head, and the upstream TGAE and pooling module. After offline training, the predicted value of each cluster is stored as its semantic potential, $\Phi_i = \widehat{\rho}_i$, and retrieved during online PBRS. To keep four quantities distinct: r_t^{env} is the environmental reward of a transition, ρ_i is the supervised target of cluster i, $\widehat{\rho}_i$ is the model prediction, and Φ_i is the potential used for shaping. Since only K scalar targets supervise the full model, we additionally report the trivial baseline $\Phi_i = \rho_i$ (no learned model) in Table 4.

3.2.5 Offline joint training of the semantic reward model

The framework performs semantic reward model optimization entirely offline before RL policy optimization. This separation avoids the instability of jointly updating the reward estimator and dialogue policy, resulting in a stationary supervised learning problem. All components are optimized end-to-end using the reward prediction objective and complementary regularization terms. Gradients propagate through the reward head, LoRA layers, ACPC, TGAE, and differentiable MinCut pooling.

Several auxiliary objectives preserve graph structure and stabilize optimization. The MinCut objective promotes strong intra-cluster and weak inter-cluster connectivity, the orthogonality constraint prevents degenerate assignments, temporal consistency encourages coherent representations of clusters at nearby dialogue positions, entropy regularization reduces uncertainty in soft assignments, and embedding regularization constrains the representation magnitude. Let π denote the ordering of clusters according to their temporal positions τ_i. The temporal consistency loss is

$$\mathcal{L}_{\text{temp}} = \frac{1}{K-1}\sum_{i=1}^{K-1} \left\| h_{\pi_{i+1}} - h_{\pi_i} \right\|_1 \tag{31}$$

where h_{π_i} is the representation of the cluster at the i-th temporal position. The embedding and entropy regularizers are

$$\mathcal{L}_{\text{emb}} = \|H\|_F^2 \tag{32}$$

$$\mathcal{L}_{\text{ent}} = -\frac{1}{NK}\sum_{i=1}^{N}\sum_{j=1}^{K} S_{ij}\log(S_{ij}) \tag{33}$$

where $S \in \mathbb{R}^{N\times K}$ is the soft assignment matrix. Minimizing $\mathcal{L}_{\text{ent}}$ encourages more confident assignments while retaining differentiable pooling; its effect on soft-assignment-based sharing is examined in Table 4. The complete offline objective is

$$\mathcal{L} = \mathcal{L}_{reward} + \lambda_1\mathcal{L}_{\text{MinCut}} + \lambda_2\mathcal{L}_{\text{orth}} + \lambda_3\mathcal{L}_{\text{temp}} + \lambda_4\mathcal{L}_{\text{ent}} + \lambda_5\mathcal{L}_{\text{emb}} \tag{34}$$

where $\lambda_1, \ldots, \lambda_5$ control the regularization terms. During offline optimization, the pretrained MiniLM backbone remains frozen, while the LoRA layers, ACPC, TGAE, pooling module, and reward head are updated:

$$\theta = \{\theta_{\text{LoRA}}, \theta_{\text{Prompt}}, \theta_{\text{Graph}}, \theta_{\text{Pool}}, \theta_{\text{Head}}\} \tag{35}$$

where θ_{LoRA}, θ_{Prompt}, θ_{Graph}, θ_{Pool}, and θ_{Head} denote the trainable parameters of LoRA, ACPC, the TGAE, the MinCut pooling module (assignment weights W_s and projection W_p, b_p), and the reward head, respectively.

3.2.6 Online reward shaping

After offline training, the assignment network and the cluster potentials are fixed, and only the dialogue policy is optimized. For each newly observed state s_t, the structural-semantic representation x_t of Section 3.2.1 is constructed and passed through the trained assignment network, Eq. (36).

$$S_t = f_{\text{assign}}(x_t) \tag{36}$$

where $S_t \in \mathbb{R}^K$ contains the assignment probabilities of s_t over the K clusters. The semantic potential is the probability-weighted combination of the stored cluster potentials, Eq. (37).

$$\Phi(s_t) = S_t^\top \Phi_c \tag{37}$$

where $\Phi_c = \{\Phi_1, \ldots, \Phi_K\}$ are the learned cluster potentials. Online, $\Phi(s_t)$ depends on the dialogue stage through the turn features contained in s_t and through the cluster-level temporal positions τ_i learned offline; temporal encoding therefore shapes the cluster potentials rather than being recomputed for each state. During online RL, the frozen MiniLM is applied once to the serialized semantic frame of each new state to obtain e_t^{sem} (its output can be cached for repeated frames), followed by a linear assignment and a K-dimensional dot product. The LoRA-adapted encoder, ACPC, and the graph encoder are not evaluated online; their effect is absorbed into Φ_c. Because assignment is feature-driven, states that are rare or absent in the offline data are mapped to related clusters through their features; the deployment-time coverage

analysis in Section 4.2.1 analyses how many deployment states fall outside the sampled graph.

The retrieved potential is incorporated into PBRS (Sang et al. 2023b; Gerasyov and Makarov 2023), Eq. (38), where the potential of terminal states is set to zero (policy invariance is established in Section 3.2.8).

$$r_t^{shape} = r_t^{env} + \beta \left(\gamma(1 - d_t)\Phi\left(s_{t+1}\right) - \Phi\left(s_t\right)\right) \tag{38}$$

where r_t^{env} is the environmental reward, γ is the discount factor, d_t indicates termination, and $\beta = 1$ (standard potential-based shaping) controls the contribution of the shaping term; potentials are used without additional normalization. Table 1 summarizes the main quantities and their dimensions used throughout the proposed approach. Fig. 3 illustrates the overall SST-RS approach. In the offline stage, collected dialogue trajectories are transformed into a semantic transition graph and compressed into a behavioral graph, where the semantic reward model learns cluster-level potentials (Fig. 4). In the online stage, each new state is softly assigned to the learned clusters, and its potential is obtained as a weighted combination of the stored cluster potentials and incorporated into PBRS (Fig. 5).

Table 1 Dimensions of the main quantities of SST-RS.

Quantity	**Symbol**	**Dimension**
Structural state vector	s_i	224
Semantic frame embedding	e_i^{sem}	384
Fused node feature	x_i	608
Node feature matrix	X	$N \times 608$
Soft assignment	S	$N \times K$
Cluster representation	Z	$K \times 608$
Pooled adjacency	A_c	$K \times K$
Projected cluster, temporal encoding	$\tilde{z}_i$, p_i	128
TGAE output	H	$K \times 128$
ACPC latent	c_i	128
Continuous prompt	Π_i	12×384
MiniLM output	u_i	384
Predicted value / potential	$\widehat{\rho}_i = \Phi_i$	scalar ($\Phi_c \in \mathbb{R}^K$)
Online assignment	S_t	K

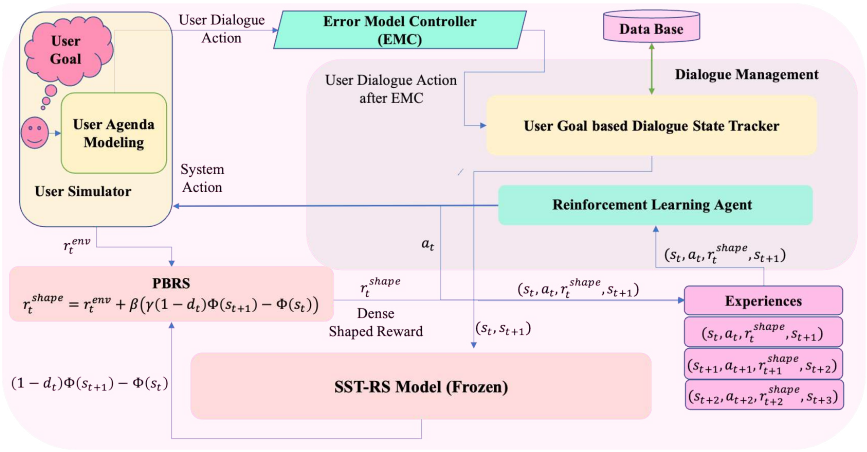


Fig. 3 Overall architecture of the dialogue system incorporating the proposed SST-RS approach for addressing reward sparsity.

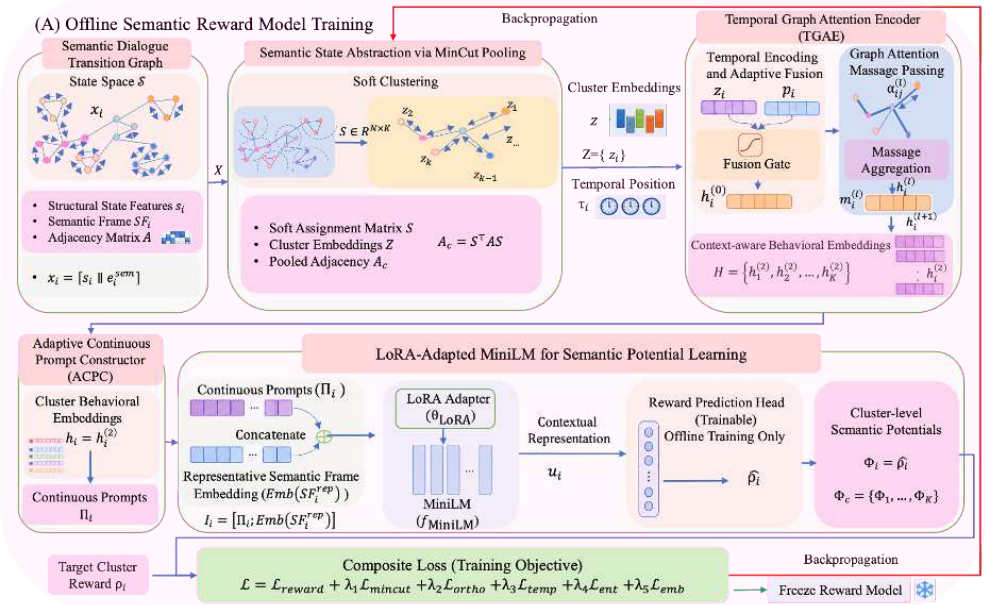


Fig. 4 Offline stage of SST-RS: semantic reward model training.

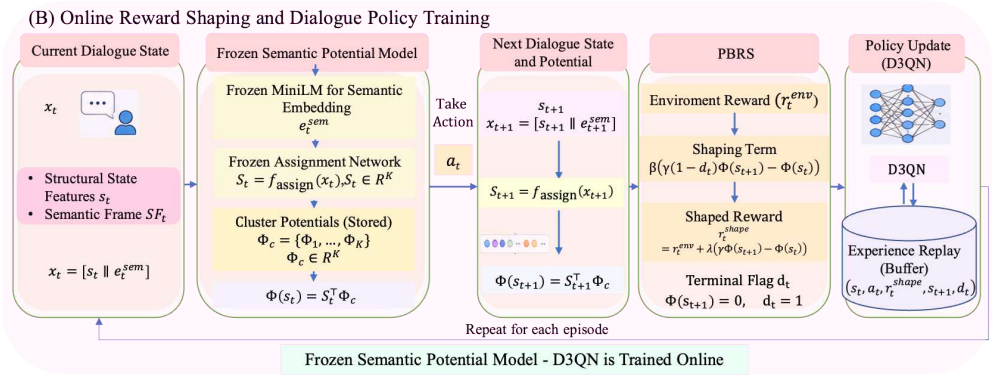


Fig. 5 Online stage of SST-RS: reward shaping with stored cluster potentials.

3.2.7 Overall optimization procedure of the proposed SST-RS approach

The overall procedure is summarized in Algorithm 1. It consists of (i) offline semantic potential learning and (ii) online dialogue policy optimization. After offline convergence, the learned clusters and potentials are fixed for online reward shaping. The

Algorithm 1 Overall Training Procedure of the Proposed SST-RS Approach

Input: Collected dialogue trajectories $\mathcal{D}$, pretrained MiniLM encoder, differentiable MinCut pooling, TGAE, ACPC, and LoRA layers.
Output: Optimized dialogue policy π^*
Offline Semantic Potential Learning
1: Construct the semantic dialogue transition graph G from $\mathcal{D}$
2: Learn soft behavioral cluster assignments using differentiable MinCut pooling
3: Generate contextual behavioral representations using the TGAE
4: Generate continuous prompts using ACPC
5: Estimate cluster-level values: $\widehat{\rho} = \text{RewardHead}(\text{LoRA-MiniLM}([\Pi; \text{Emb}(SF)]))$
6: Optimize $\mathcal{L} = \mathcal{L}_{reward} + \lambda_1 \mathcal{L}_{\text{MinCut}} + \lambda_2 \mathcal{L}_{\text{orth}} + \lambda_3 \mathcal{L}_{\text{temp}} + \lambda_4 \mathcal{L}_{\text{ent}} + \lambda_5 \mathcal{L}_{\text{emb}}$ w.r.t. θ
7: Freeze f_{assign} and store cluster potentials $\Phi_i = \widehat{\rho}_i$, $i = 1, \ldots, K$
Online Reward Shaping and Policy Optimization
8: **for** each episode **do**
9: Observe current dialogue state s_t
10: Construct structural-semantic representation x_t
11: Compute soft assignment $S_t = f_{assign}(x_t)$
12: Retrieve semantic potential $\Phi(s_t) = S_t^\top \Phi_c$
13: Observe next state s_{t+1} and compute $\Phi(s_{t+1})$
14: Compute $r_t^{shape} = r_t^{env} + \beta(\gamma(1-d_t)\Phi(s_{t+1}) - \Phi(s_t))$
15: Update D3QN using the shaped reward
16: **end for**

dialogue policy is optimized using a D3QN agent. The shaped reward is directly used to construct the temporal-difference target, while double Q-learning selects the next action with the online network and evaluates it with the target network, Eq. (39).

$$y_t = r_t^{shape} + \gamma(1-d_t)Q_{\psi^-}\left(s_{t+1}, \arg\max_{a'} Q_\psi(s_{t+1}, a')\right), \qquad \psi^- \leftarrow \kappa\psi + (1-\kappa)\psi^- \tag{39}$$

where ψ and ψ^- are the parameters of the online and target Q-networks, and κ is the target-update rate. The online network minimizes the temporal-difference loss, Eq. (40).

$$\mathcal{L}_{D3QN} = \frac{1}{B}\sum_{t=1}^{B}\left(y_t - Q_\psi(s_t, a_t)\right)^2 \tag{40}$$

where B is the minibatch size. The semantic reward model remains fixed during this stage, providing a stationary shaping signal. By decoupling potential learning from policy optimization, SST-RS avoids reward drift caused by simultaneous reward-model and policy updates, and, as shown in Section 3.2.8, the fixed-potential formulation preserves the optimal policy of the underlying MDP.

3.2.8 Policy invariance of the shaped reward

This subsection explains why the learned semantic potentials do not change the optimal dialogue policy. Before online RL starts, the assignment network f_{assign}, the frozen MiniLM used to compute e^{sem}, and the cluster potentials Φ_c are fixed. The potential $\Phi(s) = f_{\text{assign}}(x(s))^\top \Phi_c$ is therefore a fixed, deterministic function of the dialogue state s alone: it does not depend on the action or on the policy parameters, and during policy learning it is only retrieved, never updated. Together with a constant β, the same discount factor γ in the shaping term and in the learner, and $\Phi(s) = 0$ at terminal states, this is the setting of potential-based shaping (Ng et al. 1999).

Proposition 1 (policy invariance). Let M be the dialogue MDP and let M' be the same MDP with shaped reward $r'(s,a,s') = r(s,a,s') + \beta\left(\gamma\Phi(s') - \Phi(s)\right)$, where Φ is fixed, depends only on the state, and $\Phi(s) = 0$ for terminal s. Then $Q^*_{M'}(s,a) = Q^*_M(s,a) - \beta\Phi(s)$ for all (s,a). Consequently $\arg\max_a Q^*_{M'}(s,a) = \arg\max_a Q^*_M(s,a)$ for every state, and a policy is optimal in M' if and only if it is optimal in M.

Proof. Define $\tilde{Q}(s,a) = Q^*_M(s,a) - \beta\Phi(s)$. By the Bellman optimality equation in M,

$$Q^*_M(s,a) = \mathbb{E}_{s'}\left[r(s,a,s') + \gamma\max_{a'} Q^*_M(s',a')\right]. \tag{41}$$

Subtracting $\beta\Phi(s)$ on both sides, and adding and subtracting $\beta\gamma\Phi(s')$ inside the expectation, gives

$$\begin{aligned}\tilde{Q}(s,a) &= \mathbb{E}_{s'}\left[r(s,a,s') + \beta\big(\gamma\Phi(s') - \Phi(s)\big) + \gamma\max_{a'}\left(Q^*_M(s',a') - \beta\Phi(s')\right)\right] \\ &= \mathbb{E}_{s'}\left[r'(s,a,s') + \gamma\max_{a'}\tilde{Q}(s',a')\right],\end{aligned} \tag{42}$$

where $\max_{a'}$ passes through the term $\beta\Phi(s')$ because Φ does not depend on a'. Hence $\tilde{Q}$ satisfies the Bellman optimality equation of M', whose fixed point is unique, so

$Q^*_{M'} = \tilde{Q}$. At terminal states $Q^*_M(s', \cdot) = 0$ and $\Phi(s') = 0$, so the relation holds there as well. Since $\beta\Phi(s)$ does not depend on a, the maximizing actions coincide.

Three remarks qualify this result. First, it requires Φ to be a function of the tracked dialogue state only; the state used by the potential is the tracker state together with the semantic frames from which it is built, and $\Phi(s)$ is a deterministic function of this state. Second, invariance is a statement about the optimal policy of the underlying MDP. With function approximation, finite exploration, and ϵ-greedy action selection, the potentials change the learning dynamics, which is what the experiments evaluate, and a poorly estimated potential can slow rather than speed up learning. Third, the result relies on Φ being frozen; updating Φ during RL would require the analysis of dynamic potential-based shaping and is not done here. Since $G^{shaped}(\mathcal{T}) = G(\mathcal{T}) - \beta\Phi(s_0)$ when $\Phi = 0$ at the terminal state (the shaping terms telescope), the success rate and the number of turns, which are computed from the environment outcome rather than from the shaping term, are not offset by shaping, whereas the reported average reward is the shaped return and should be read with this offset in mind.

4 Experimental results and discussion

Experiments evaluate the effectiveness of the proposed approach in alleviating reward sparsity in RL-based TOD systems. First, ablation studies examine the contribution of individual components. The approach is then compared with state-of-the-art dense-reward and DPL methods.

4.1 Experimental setup and evaluation protocol

4.1.1 Datasets

The experiments use benchmark TOD environments from the Microsoft Dialogue Challenge (Li et al. 2017; Wang et al. 2020a). These environments provide rule-based user simulators and, optionally, pretrained NLU/NLG modules; in this work the agent interacts with the simulator at the semantic-frame level, so NLU and NLG are not used. The dialogue corpora were originally collected through Amazon Mechanical Turk and subsequently refined and validated by domain experts. Experiments are conducted on three TOD domains: Movie Ticket Booking, Restaurant Reservation, and Taxi Ordering. Each domain defines a specific dialogue goal with its corresponding intents and slots. The statistics of the three domains are summarized in Table 2. Each dialogue state contains inform slots, which represent user constraints, and request slots,

Table 2 Statistics related to the datasets.

Domain	Intents	Slots	Dialogues
Movie ticket booking	11	29	2890
Restaurant reservation	11	30	4103
Taxi ordering	11	29	3094

which specify the information requested from the system. At the beginning of an episode, the user simulator initializes a dialogue goal, and the policy interacts with the simulator to complete the task (Li et al. 2017).

4.1.2 Experimental settings

All experiments were implemented in Python 3.10 using PyTorch, PyTorch Geometric, and HuggingFace Transformers. Behavioral graph construction used NetworkX, and sparse graph operations used Torch-Sparse. Experiments were conducted on a MacBook M1 with an 8-core CPU and 8 GB unified memory.

Following previous TOD studies (Li et al. 2017; Saffari et al. 2025), all dialogue policies were implemented using a D3QN agent with a two-layer MLP (80 hidden units). Models were trained for approximately 9,000 episodes using a replay buffer of 10,000 transitions, a mini-batch size of 32, and a target network updated by exponential moving average with $\kappa = 0.01$. An ϵ-greedy exploration strategy was adopted ($\epsilon = 0.2$, decay $= 0.95$), together with a 5% slot error rate, discount factor $\gamma = 0.9$, and a 1,000-transition warm-up phase. Policy optimization used RMSprop with a learning rate of 10^{-3}. The policy receives the 224-dimensional state vector directly. The shaping weight is $\beta = 1$ (standard potential-based shaping), potentials are used without additional normalization, and the soft cluster assignment of Eq. (36) is used at deployment as in training. Reported metrics are measured at the end of the 9,000 training episodes. Baseline reward formulations and hyperparameters follow the original publications.

The behavioral graph was constructed from trajectories collected by a DQN agent under the original sparse reward setting. 100,000 transitions were collected for Movie Ticket Booking (after the success rate exceeded 80%), and 10,000 transitions for Restaurant Reservation and Taxi Ordering (after reaching 30% success). The thresholds and transition budgets were chosen empirically: Restaurant Reservation and Taxi Ordering are harder under sparse rewards, so a lower success threshold and a smaller transition budget were used for them. The maximum dialogue length was set to 40 turns for Movie Ticket Booking and 30 turns for the other domains. Although the graph was built from DQN trajectories, all comparative experiments used a D3QN agent (Saffari et al. 2025).

Each dialogue state was represented by the concatenation of its structural state representation and a semantic embedding from the frozen MiniLM-L12-H384-uncased encoder. Following Saffari et al. (2025), the number of behavioral clusters is $K = 30$. The graph was compressed using a differentiable MinCut pooling layer with a single linear assignment network followed by Softmax and a linear projection layer.

The reward model was optimized offline using AdamW with a learning rate of 10^{-4}, weight decay of 10^{-5}, gradient clipping with maximum norm 1.0, and 200 epochs. The MiniLM backbone remained frozen; following Hu et al. (2022), LoRA adapters ($r = 4$, $\alpha_{\text{LoRA}} = 8$, dropout 0.05) were injected into the query and value projection matrices of all 12 transformer layers. Only these adapters, together with the pooling module, the TGAE, the ACPC, and the reward head, were updated.

The TGAE consisted of two graph message-passing layers with learnable attention and hidden size $d_h = 128$ (the 608-dimensional cluster representation is linearly

projected to d_h by the pooling module). Temporal positions were encoded with sinusoidal positional encoding and fused with behavioral representations through learnable gates. Messages were propagated over the symmetrically normalized behavioral graph, followed by gating, residual connections, layer normalization, GELU, and dropout ($p = 0.1$). The ACPC was a two-layer MLP with GELU and dropout ($p = 0.1$) with a 128-dimensional hidden layer generating $L = 12$ continuous prompt tokens of dimension $d_e = 384$, which were prepended to the semantic-frame token embeddings. The reward head was a two-layer MLP with 128 hidden units, ReLU, dropout ($p = 0.1$), and a linear output. The loss coefficients $(\lambda_1, \ldots, \lambda_5)$ were set to $(0.1, 0.1, 0.01, 0.01, 0.001)$ and kept fixed across domains.

4.1.3 State-of-the-art baseline models

SST-RS addresses reward sparsity through an offline-trained semantic reward model and PBRS; the baselines are therefore considered from two perspectives: dense reward and reward-shaping methods, and dialogue policy optimization methods.

1) Dense Reward Construction Methods: These baselines alleviate sparse rewards by introducing additional supervisory information or intrinsic rewards. For fairness, the original reward configuration of each baseline is preserved, while all other settings are standardized; all are evaluated under the same D3QN policy, the same training budget (9,000 episodes), the same seeds, and the same offline dataset where a method uses one.

- **GAT+MinCut+CSR** (Saffari et al. 2025): constructs a dialogue graph from structural state features, applies MinCutPool for soft clustering, and uses a Graph Attention Network to learn neighborhood-aware representations for cluster-based reward estimation. Unlike the proposed method, it does not incorporate semantic-frame features into reward estimation.
- **S2Agent** (Wang et al. 2020a): combines policy shaping and reward shaping through imitation learning; the imitation model is trained on the same collected dialogue dataset.
- **LHUA** (Lu et al. 2021): employs hindsight-based learning to extract supervision from unsuccessful trajectories by reformulating dialogue goals; hindsight experiences are drawn from the same dataset.
- **FeudalGain** (Geishauser 2024): introduces an intrinsic reward based on dialogue information gain.
- **MACRM** (Sun et al. 2025): combines environmental rewards with curiosity-driven intrinsic rewards; the single-agent variant is adopted with the original weighting schedule.

2) Dialogue Policy Optimization Methods: These methods use different policy architectures and reward formulations, and their results are taken from the original papers.

- **GSDP** (Xu et al. 2024): represents dialogue states using structured bipartite graphs and combines graph convolution with recurrent modeling to improve dialogue policy optimization.

- **MGPL** (Liang et al. 2025): adopts a multi-agent actor-critic framework with alternating policy optimization, enabling coordinated exploration and more efficient policy learning.
- **MCA** (Xu et al. 2025): adopts a model-based RL framework that generates dialogue trajectories using an autoregressive Transformer and incorporates causal modeling of user intentions for policy optimization.

4.1.4 Measurement metrics

Three metrics are employed: success rate, average cumulative reward, and average dialogue turns. The success rate is the ratio of successfully completed dialogues to the total number of episodes. A dialogue is successful when the agent enables the user simulator to achieve the specified goal while satisfying the required constraints. The average reward is the mean cumulative reward over a specified number of episodes, computed from the shaped reward used for training (Eq. (38)); because the shaping term telescopes, this return differs from the environmental return by $-\beta\Phi(s_0)$, so the success rate and the dialogue length are the primary indicators of policy quality. The average number of dialogue turns measures interaction efficiency, where a lower value indicates a more concise dialogue. To ensure reliable evaluation, all results are averaged over five independent runs with different random seeds and reported as means over the runs. To assess the statistical significance of the difference between SST-RS and the most closely related method, paired one-tailed t-tests ($df = 4$) are conducted on the results of the five matched random seeds, with Holm–Bonferroni correction over the three metrics ($\alpha = 0.05$).

4.2 Results and discussion

This section presents the experimental evaluation. First, ablation studies investigate the contribution of each component and the learning behavior of the agent. Then, SST-RS is compared with dense-reward methods under the same protocol, and separately with previously reported results of other policy optimization methods.

4.2.1 Ablation study

An ablation study was conducted by removing or replacing one component at a time. Since differentiable MinCut pooling provides the structural abstraction backbone, it was retained in all variants of this study (its own contribution is examined in Table 5). For the temporal ablation, only the temporal encoding was removed, while message passing was preserved. Results are reported in Table 3. **Effect of Semantic Frames.** Removing semantic frame representations causes the largest degradation, reducing the success rate from 94.00% to 88.40%, the average reward from 29.10 to 23.43, and increasing the average dialogue length from 7.02 to 8.82 turns. Although structural state features capture slot status, they provide limited information about relationships between behaviorally similar states; the MiniLM-based representations enrich the graph with contextual information. The result shows that the information carried by the frame representation is important for estimating informative potentials.

Table 3 Ablation study of the proposed approach for the Movie ticket booking task. Metrics include success rate (%) (Success), average reward (Reward), average turns (Turn).

Model	Success	Reward	Turns
Sparse reward	82.40	18.63	10.16
w/o Semantic Frames	88.40	23.43	8.82
w/o Temporal Behavioral Encoding	89.60	25.56	8.48
w/o Adaptive Continuous Prompting	91.60	27.37	7.98
SST-RS (Full Model)	**94.00**	**29.10**	**7.02**

Effect of Temporal Behavioral Encoding. Removing temporal behavioral encoding reduces the success rate to 89.60%, the reward to 25.56, and increases the length to 8.48 turns. Since message passing is unchanged, this suggests that graph topology alone does not represent the evolution of dialogue behavior; temporal encoding helps distinguish behavioral regions that occur at different dialogue stages.

Effect of Adaptive Continuous Prompting. Replacing ACPC with a direct feature projection decreases the success rate from 94.00% to 91.60%, reduces the reward to 27.37, and increases the length to 7.98 turns. As the graph and temporal modeling are unchanged, the degradation is attributed to a weaker interaction between cluster representations and the MiniLM encoder.

Removing any single component lowers the success rate to between 88.40% and 91.60%, below the 94.00% of the complete approach, and it also lowers the reward and lengthens the dialogues. The size of the degradation follows the order semantic frames, temporal encoding, and ACPC, which indicates that the components contribute in a complementary way rather than through a single module.

Compared with sparse-reward training, the full SST-RS increases task success from 82.40% to 94.00% (+11.60 points), increases the average reward from 18.63 to 29.10, and reduces the average dialogue length from 10.16 to 7.02 turns.

Differentiable MinCut pooling compresses the dialogue-state graph into compact behavioral clusters and enables cluster-level potentials instead of state-specific rewards, which reduces state redundancy. The use of a lightweight MiniLM with LoRA is intended to keep semantic modeling cheaper than LLM-based approaches. Since the reward model is trained offline and cluster potentials are reused, repeated adaptation of the language model during policy optimization is avoided.

Table 4 isolates the role of the potential definition. A random fixed potential yields essentially no improvement over sparse-reward training (83.20% versus 82.40%), so the gains of SST-RS do not stem from adding an arbitrary state-dependent shaping term. Using the cluster-mean target directly as the potential, without the learned reward model, reaches 90.80% success with a reward of 25.97 and 7.90 turns, i.e., it recovers about 72% of the 11.60-point success gain of the complete approach. This indicates that most of the benefit originates from the MinCut abstraction and from sharing reward information across behaviorally related states. The learned semantic reward model accounts for the remaining 3.20 points together with higher reward and shorter dialogues, which is the role of the semantic-frame, temporal, and prompt-conditioned

components analyzed above. Replacing the mean-reward target by the discounted return-to-go gives a similar result (93.20%), slightly below the complete approach; this agrees with Proposition 1, which implies that the choice of target affects the speed and stability of learning rather than the optimal policy, and it suggests that the simpler mean-reward target is adequate in these domains.

Table 4 Potential-definition controls on Movie ticket booking (mean over five seeds).

Variant	**Success**	**Reward**	**Turns**
Random fixed potential	83.20	19.45	9.92
Cluster mean, $\Phi_i = \rho_i$ (no learned model)	90.80	25.97	7.90
Return-to-go target	93.20	28.68	7.57
SST-RS (Full Model)	94.00	29.10	7.02

Table 5 examines structural choices and training components. Replacing MinCut by k-means lowers the success rate from 94.00% to 89.20% and the reward from 29.10 to 25.41, and it lengthens the dialogues from 7.02 to 8.26 turns, so the learned, structure-aware soft assignment contributes beyond feature-space clustering at the same number of clusters. For temporal modeling, positional encoding without message passing (90.40%) and message passing without temporal encoding (89.60%) are both clearly below the complete encoder, which indicates that temporal and relational information are complementary. For the language encoder, training only the prompts on a frozen MiniLM reaches 92.40%, and LoRA adds 1.60 points (94.00%). Full fine-tuning gives a nearly identical 94.40% while updating all roughly 33 million encoder parameters instead of about 74 thousand low-rank adapter parameters (rank 4 on the query and value projections), which supports parameter-efficient adaptation. Finally, removing the temporal-consistency, entropy, or embedding regularizer reduces the success rate by 0.40 to 1.00 points and worsens reward and dialogue length, so these terms have a small but consistent positive effect.

The sensitivity analysis (Table 6) shows that performance is highest at $K = 30$ (94.00%) and decreases for both a coarser ($K = 10$, 90.40%) and a finer partition ($K = 50$, 93.00%), plausibly because a coarse partition merges behaviorally different states whereas a fine one fragments the reward information. The prompt length and the LoRA rank have little influence in the tested ranges (92.40–94.00% for L between 4 and 16, and 93.20–94.40% for r between 2 and 16), so these parameters do not require careful tuning. The shaping weight is the most sensitive parameter: $\beta = 1$ (standard PBRS) performs best, whereas a large weight ($\beta = 4$) reduces the success rate to 89.60%, since the potential then dominates the environmental reward.

Table 7 compares SST-RS with GAT+MinCut+CSR (Saffari et al. 2025), the most closely related method, over the five matched seeds. SST-RS is better than GAT+MinCut+CSR in all five matched runs on each of the three metrics, and the differences are statistically significant after Holm–Bonferroni correction (adjusted $p = 0.006$ for success rate and 0.035 for both reward and dialogue length). The effect

Table 5 Structure, temporal modeling, adaptation, and loss ablations on Movie ticket booking (mean over five seeds).

Variant	**Success**	**Reward**	**Turns**
Clustering			
k-means clusters instead of MinCut	89.20	25.41	8.26
Temporal modeling			
Positional encoding only (no message passing)	90.40	26.38	8.23
Message passing only (w/o temporal encoding)	89.60	25.56	8.48
TGAE (full)	94.00	29.10	7.02
Adaptation of MiniLM			
Frozen MiniLM, prompts only	92.40	27.62	7.84
Full fine-tuning	94.40	29.31	7.35
LoRA (ours)	94.00	29.10	7.02
Auxiliary losses			
w/o $\mathcal{L}_{\text{temp}}$	93.00	28.38	7.56
w/o $\mathcal{L}_{\text{ent}}$	93.20	28.54	7.51
w/o $\mathcal{L}_{\text{emb}}$	93.60	28.87	7.44

Table 6 Sensitivity of the success rate (%) on Movie ticket booking to the main hyperparameters (mean over five seeds).

Parameter	**Value 1**	**Value 2**	**Value 3**	**Value 4**
Clusters $K \in \{10, 20, 30, 50\}$	90.40	92.80	94.00	93.00
Prompt length $L \in \{4, 8, 12, 16\}$	92.40	93.20	94.00	93.60
LoRA rank $r \in \{2, 4, 8, 16\}$	93.20	94.00	94.40	93.60
Shaping weight $\beta \in \{0.5, 1, 2, 4\}$	93.00	94.00	92.80	89.60

sizes are large because the paired differences are consistent across runs, but the absolute differences are small (1.20 points in success rate, 0.95 in reward, and 0.04 turns), so the practical gain over the prior method is modest, in particular for the dialogue length. The margins over the other dense-reward baselines are larger (3.60 to 6.00 points in success rate, Table 9).

Table 7 Statistical comparison of SST-RS with GAT+MinCut+CSR on Movie ticket booking over the five matched random seeds (paired one-tailed t-test in the direction of improvement, $df = 4$; p is Holm–Bonferroni adjusted over the three metrics). Δ is the mean paired difference (SST-RS minus GAT+MinCut+CSR).

Metric	Δ	t	p **(Holm)**	**Cohen's** d_z
Success rate (%)	1.20	6.00	0.006	2.68
Average reward	0.95	3.15	0.035	1.41
Average turns	-0.04	-2.65	0.035	-1.19

To examine the consistency of the gains across dialogue tasks, the experiments were repeated on the Restaurant Reservation and Taxi Ordering domains; a separate model is trained for each domain, so this is a multi-domain evaluation rather than a transfer experiment. Only the success rate is reported (Table 8). SST-RS achieves the

Table 8 Task success rate (%) of SST-RS and its ablations on the three domains (a separate model is trained for each domain).

Model	Movie	Restaurant	Taxi
Sparse reward	82.40	53.20	79.00
w/o Semantic Frames	88.40	55.40	84.00
w/o Temporal Behavioral Encoding	89.60	57.00	86.80
w/o Adaptive Continuous Prompting	91.60	57.60	88.40
SST-RS (Full Model)	**94.00**	**59.60**	**91.60**

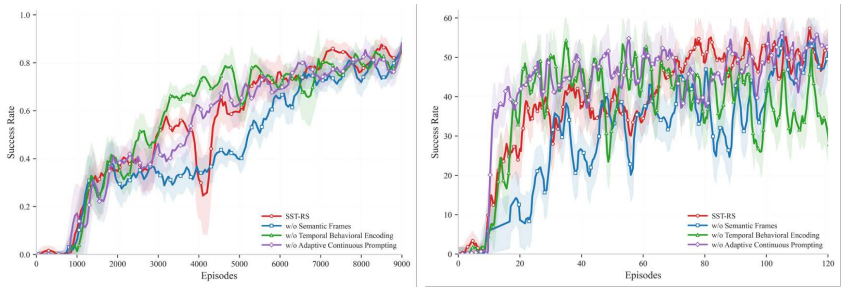
highest success rate in all three domains. Compared with sparse-reward training, it improves the success rate by 11.60, 6.40, and 12.60 points on Movie Ticket Booking, Restaurant Reservation, and Taxi Ordering, respectively (mean 10.20 points). Movie Ticket Booking achieves the highest overall performance, whereas Restaurant Reservation remains the most challenging, consistent with its greater behavioral diversity and constraint combinations. In Restaurant Reservation, the contribution of the individual components is smaller in absolute terms (2.00 points between the full model and the variant without ACPC), in line with the lower overall success rates of this domain.

The ordering of the ablation variants is largely consistent across domains: removing semantic representations produces the largest degradation, followed by removing temporal encoding, while removing ACPC produces the smallest decrease. By representing groups of related states rather than individual states, SST-RS can provide reward information beyond frequently visited states.

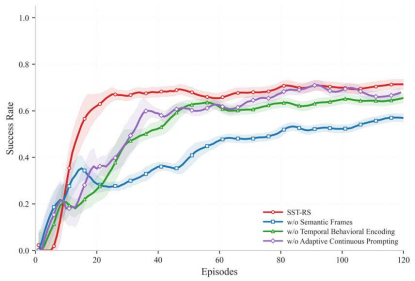
Deployment-time state coverage.

To examine whether the states met during policy optimization differ from those used to build the graph, the states encountered by the D3QN agent were compared with the sampled states used for graph construction on Movie Ticket Booking. After excluding the three continuous-valued features (indices 123, 132, and 148), none of the 34,192 unique deployment states exactly matched any of the 13,649 unique states of the original DQN set (0% exact overlap), whereas 64.85% of the deployment states shared at least 90% of their features with some original state. The deployment process therefore regularly encounters states that were not directly observed during graph construction. For such states, including isolated ones, the feature-driven soft assignment provides a mapping to the learned clusters and hence a potential, without requiring state-level neighbors. Since independent ground-truth values are not available for unseen states, this analysis does not quantify the accuracy of the transferred potentials.

To investigate the learning behavior, Fig. 6 presents the success-rate learning curves across the three domains.



(a) Movie Ticket Booking **(b)** Restaurant Reservation



(c) Taxi Ordering

Fig. 6 Learning curves of the average success rate on the Movie Ticket Booking, Restaurant Reservation, and Taxi Ordering tasks.

As illustrated in Fig. 6(a), SST-RS shows superior overall learning behavior and converges to the highest success rate on Movie Ticket Booking. During early exploration, all methods perform similarly because updates rely mainly on sparse environmental rewards. As training proceeds, the semantic potentials provide increasingly informative intermediate feedback, accelerating policy improvement and reducing fluctuations. Although some ablation variants temporarily achieve comparable or slightly higher success rates in intermediate stages, their trajectories fluctuate more and converge lower. Removing semantic frame embeddings causes the largest degradation, while eliminating temporal encoding or ACPC also slows convergence.

A more challenging pattern appears in Restaurant Reservation (Fig. 6(b)). The larger behavioral diversity leads to greater fluctuations across methods. Nevertheless, SST-RS converges to the highest final success rate with a more stable later-stage optimization. Removing semantic frames results in a clear degradation, suggesting that contextual semantic information becomes valuable when trajectories with similar structure correspond to different user constraints. The variant without ACPC temporarily reaches or slightly exceeds the full approach in early and intermediate stages, but saturates below it. The temporal ablation also exhibits less stable learning.

In Taxi Ordering (Fig. 6(c)), SST-RS learns faster in the early stages, reaches a high-performance region earlier, and converges to the highest success rate of 91.60%. The variant without ACPC gradually approaches the full approach but remains

below it, and removing temporal encoding leads to slower convergence. Consistent with Tables 3 and 8, removing semantic frames again produces the weakest learning behavior. Compared with Restaurant Reservation, Taxi Ordering exhibits smoother optimization and smaller fluctuations. Taken together, the curves reveal domain-specific convergence patterns with a consistent advantage for SST-RS.

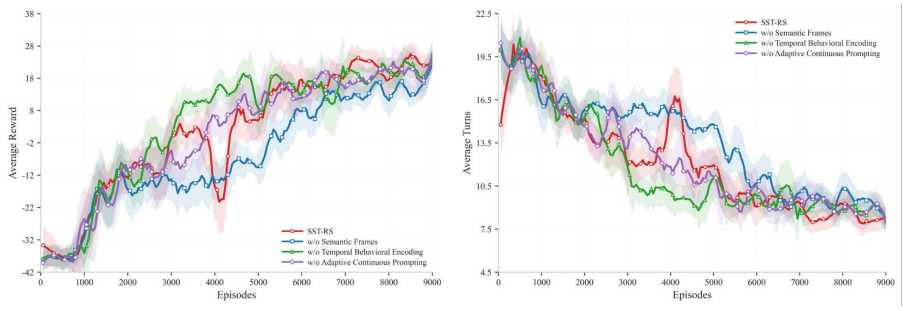
Fig. 7 presents the learning curves of the average accumulated reward and the average dialogue length on Movie Ticket Booking. As shown in Fig. 7(a), SST-RS converges to the highest accumulated reward. The advantage becomes evident as training progresses, indicating that the learned potentials provide more informative shaping signals. Some ablation variants temporarily obtain comparable rewards in intermediate stages but with larger oscillations and lower long-term stability. Removing semantic frames leads to the largest reward degradation, while removing temporal encoding or ACPC also reduces the reward.

Dialogue length reflects policy efficiency. As shown in Fig. 7(b), all methods initially require more turns because the policy is still exploring; with training, dialogue length decreases. SST-RS converges to the shortest dialogue length in the final stages while achieving the highest success rate and reward. Removing semantic representations produces the largest increase in dialogue length, followed by the temporal and ACPC ablations.

Overall, the learning curves are consistent with the quantitative results of Tables 3 and 8: SST-RS achieves higher success, larger reward, and shorter dialogues than the sparse-reward baseline and all ablation variants. These results suggest that the framework can provide reusable dense rewards without extensive human supervision or online LoRA/prompt inference, although the claims on scalability to larger state spaces remain to be tested.

4.2.2 Comparison with state-of-the-art baselines

SST-RS is compared with representative dense-reward methods and the graph-based method GAT+MinCut+CSR (Saffari et al. 2025). For fairness, all dense-reward approaches are evaluated with the same D3QN policy, retaining their respective



(a) Average Reward **(b)** Average Turns

Fig. 7 Learning curves of the average accumulated reward and the average dialogue length during training on the Movie Ticket Booking domain.

reward-generation mechanisms (Table 9). Policy optimization methods that use different architectures, simulators, or reward formulations are listed separately as reported results and are not directly comparable (Table 10). Evaluation is on Movie Ticket Booking after 9,000 training episodes. As shown in Table 9, SST-RS achieves the

Table 9 Comparison with dense-reward methods under the same D3QN policy and protocol on Movie Ticket Booking.

Model	Success	Reward	Turns
S2Agent (Wang et al. 2020a)	88.00	23.56	9.38
LHUA (Lu et al. 2021)	89.00	23.72	7.92
FeudalGain (Geishauser 2024)	90.40	19.50	8.14
MACRM (Sun et al. 2025)	88.80	24.48	7.96
GAT+MinCut+CSR (Saffari et al. 2025)	92.80	28.15	7.06
SST-RS	**94.00**	**29.10**	**7.02**

Table 10 Previously reported results of other policy optimization methods on Movie Ticket Booking. Not directly comparable (different policy architectures, reward formulations, and settings).

Model (as reported)	Success	Reward	Turns
GSDP (Xu et al. 2024)	72.27	38.08	20.37
MGPL (Liang et al. 2025)	81.60	52.54	12.74
MCA (Xu et al. 2025)	90.65	58.47	–

highest success rate (94.00%) and average reward (29.10) among the methods evaluated under the same protocol, and the shortest dialogues (7.02 turns), although the dialogue length is essentially equal to that of GAT+MinCut+CSR (7.06 turns). The success gains over S2Agent, LHUA, FeudalGain, MACRM, and GAT+MinCut+CSR are 6.00, 5.00, 3.60, 5.20, and 1.20 points, respectively. The reported values of Table 10 are lower in success rate than SST-RS, whereas their rewards are on different scales because of different reward definitions; because of the differing settings this should be taken as context rather than as a direct ranking.

S2Agent and LHUA improve exploration using demonstration knowledge and hindsight experience, respectively, but derive reward signals directly from observed trajectories, which may limit their use for infrequently visited states. FeudalGain and MACRM rely on intrinsic signals such as information gain and prediction error, which do not explicitly characterize the semantic meaning or temporal contribution of a state to task completion.

The comparison with GAT+MinCut+CSR is particularly relevant since both use differentiable MinCut pooling. GAT+MinCut+CSR describes each node solely by the

224-dimensional state vector, i.e., by slot and turn features, whereas SST-RS additionally derives the meaning of each state from its semantic frame (intents, acts, slot values) and extends the representation with temporal modeling, and uses ACPC to condition a LoRA-adapted MiniLM encoder. SST-RS improves the success rate from 92.80% to 94.00% and the average reward from 28.15 to 29.10 with a comparable dialogue length. The difference is modest in absolute terms, but it is observed in all five matched runs and remains statistically significant after correction (Table 7), which is compatible with the view that enriching graph-based abstraction with semantic and temporal information can yield more informative reward signals.

5 Conclusion

This paper presented SST-RS, a semantic potential learning framework for addressing reward sparsity in RL-based TOD systems. The approach learns cluster-level semantic potentials from compressed behavioral graphs during an offline stage and incorporates them into PBRS to provide intermediate feedback during online policy optimization. By integrating differentiable MinCut-based behavioral abstraction, semantic frame representations, temporal behavioral modeling, and adaptive language-guided conditioning, SST-RS captures complementary structural, semantic, and temporal characteristics of dialogue behavior.

Experiments on three benchmark domains show that SST-RS improves task success while maintaining competitive dialogue efficiency, and the ablations indicate that semantic representations, temporal encoding, and adaptive continuous prompting each contribute. Learning cluster-level potentials, rather than independent estimates for individual states, compresses the state space into compact behavioral representations, and the lightweight MiniLM with LoRA provides semantic modeling without the overhead of large language models. Since the reward model is trained offline and the potentials are reused online, SST-RS avoids online LoRA/prompt inference and large-language-model calls during policy optimization.

Future work can investigate adaptive potentials that account for dialogue stages, user behaviors, and changing task objectives; evaluate generalization to larger multi-domain and open-domain environments and LLM-based user simulators; and examine more efficient language encoders.

Declarations

- **Funding:** The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.
- **Competing interests:** The authors have no relevant financial or non-financial interests to disclose.
- **Ethics approval and consent to participate:** Not applicable.
- **Consent for publication:** Not applicable.
- **Data availability:** The datasets used in this study are publicly available. The dialogue samples used in the experiments were extracted from these datasets, and the graph construction procedure is described in the manuscript to facilitate reproducibility.

- **Materials availability:** Not applicable.
- **Code availability:** The code implemented in this study is available upon reasonable requestThe code implemented in this study is available upon reasonable request.

References

Algherairy A, Ahmed M (2024) A review of dialogue systems: Current trends and future directions. Neural Computing and Applications 36:6325–6351. URL https://doi.org/10.1007/s00521-023-09322-1

Arjona-Medina JA, Gillhofer M, Widrich M, et al (2019) RUDDER: Return decomposition for delayed rewards. In: Advances in Neural Information Processing Systems

Bianchi FM, Grattarola D, Alippi C (2019) Mincut pooling in graph neural networks. arXiv preprint arXiv:190700481 5. URL https://openreview.net/forum?id=BkxfshNYwB

Bianchi FM, Grattarola D, Alippi C (2020) Spectral clustering with graph neural networks for graph pooling. In: International Conference on Machine Learning, pp 874–883, URL https://proceedings.mlr.press/v119/bianchi20a.html

Chan AJ, Sun H, Holt S, et al (2024) Dense reward for free in reinforcement learning from human feedback. arXiv preprint arXiv:240200782 URL https://doi.org/10.48550/arXiv.2402.00782

Chen H, Liu X, Yin D, et al (2017) A survey on dialogue systems: Recent advances and new frontiers. ACM SIGKDD Explorations Newsletter 19:25–35. URL https://doi.org/10.48550/arXiv.1711.01731

Chu Z, Wang N, Wang H (2024) Multi-objective intrinsic reward learning for conversational recommender systems. In: Advances in Neural Information Processing Systems, URL https://doi.org/10.52202/075280-0777

Durugkar I, Tec M, Niekum S, et al (2021) Adversarial intrinsic motivation for reinforcement learning. Advances in Neural Information Processing Systems 34:8622–8636. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/486c0401c56bf7ec2daa9eba58907da9-Abstract.html

El Asri L, Laroche R, Pietquin O (2013) Reward shaping for statistical optimisation of dialogue management. In: International Conference on Statistical Language and Speech Processing, pp 93–101, URL https://doi.org/10.1007/978-3-642-39593-2_8

Fan Q, Han H, Xing J, et al (2026) Adaptive reinforcement learning for recommendation via large language models and knowledge graphs. Journal of King Saud University - Computer and Information Sciences 38:345. URL https://link.springer.

com/article/10.1007/s44443-026-00745-z

Feng Y, Yang S, Zhang S, et al (2023) Fantastic rewards and how to tame them: A case study on reward learning for task-oriented dialogue systems. arXiv preprint arXiv:230210342 URL https://doi.org/10.48550/arXiv.2302.10342

Ferreira E, Lefevre F (2013) Social signal and user adaptation in reinforcement learning-based dialogue management. In: Proceedings of the 2nd Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Perception, Action and Communication, pp 61–69, URL https://doi.org/10.1145/2493525.2493535

Geishauser C (2024) Dense rewards and continual reinforcement learning for task-oriented dialogue policies. PhD thesis, Heinrich-Heine-Universität Düsseldorf, URL https://docserv.uni-duesseldorf.de/servlets/DerivateServlet/Derivate-70958/final%20PhD%20thesis%2023.%20April%202024.pdf

Gerasyov M, Makarov I (2023) Dealing with sparse rewards using graph neural networks. IEEE Access 11:89180–89187. URL https://doi.org/10.1109/ACCESS.2023.3305927

Gong Z, Fang X (2026) Activity recommendation in business process modeling with dynamic graph neural network. Engineering Applications of Artificial Intelligence 178:115044. URL https://doi.org/10.1016/j.engappai.2026.115044

Gordon-Hall G, Gorinski PJ, Lampouras G, et al (2020) Show us the way: Learning to manage dialog from demonstrations. arXiv preprint arXiv:200408114 URL https://doi.org/10.48550/arXiv.2004.08114

Hou Z, Liu B, Zhao R, et al (2021) Imperfect also deserves reward: Multi-level and sequential reward modeling for better dialog management. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 2993–3001, URL https://doi.org/10.18653/v1/2021.naacl-main.238

Hu EJ, Shen Y, Wallis P, et al (2022) Lora: Low-rank adaptation of large language models. In: International Conference on Learning Representations, p 3, URL https://doi.org/10.48550/arXiv.2106.09685

Jang Y, Lee J, Kim KE (2022) Gpt-critic: Offline reinforcement learning for end-to-end task-oriented dialogue systems. In: International Conference on Learning Representations, URL https://openreview.net/forum?id=qaxhBG1UUaS

Kamuni N, Shah H, Chintala S, et al (2024) Enhancing end-to-end multi-task dialogue systems: A study on intrinsic motivation reinforcement learning algorithms for improved training and adaptability. In: 2024 IEEE 18th International Conference on Semantic Computing (ICSC), pp 335–340, URL https://doi.org/10.1109/

ICSC59802.2024.00063

Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:160902907 URL https://doi.org/10.48550/arXiv.1609.02907

Klissarov M, Precup D (2020) Reward propagation using graph convolutional networks. In: Advances in Neural Information Processing Systems, pp 12895–12908, URL https://proceedings.neurips.cc/paper/2020/hash/970627414218ccff3497cb7a784288f5-Abstract.html

Lester B, Al-Rfou R, Constant N (2021) The power of scale for parameter-efficient prompt tuning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp 3045–3059, URL https://doi.org/10.18653/v1/2021.emnlp-main.243

Li L, Walsh TJ, Littman ML (2006) Towards a unified theory of state abstraction for MDPs. AI&M 1(2):3

Li X, Chen YN, Li L, et al (2017) End-to-end task-completion neural dialogue systems. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp 733–743, URL https://doi.org/10.48550/arXiv.1703.01008

Li XL, Liang P (2021) Prefix-tuning: Optimizing continuous prompts for generation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp 4582–4597, URL https://doi.org/10.18653/v1/2021.acl-long.353

Liang S, Xu K, Dong Z (2025) A multi-agent approach to modeling task-oriented dialog policy learning. IEEE Access 13:11754–11764. URL https://doi.org/10.1109/ACCESS.2025.3529469

Liu X, Ji K, Fu Y, et al (2022) P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp 61–68, URL https://doi.org/10.18653/v1/2022.acl-short.8

Lu J, Tang C, Ma W, et al (2025) Graph-based reinforcement learning for software-defined networking traffic engineering. Journal of King Saud University - Computer and Information Sciences 37:119. URL https://link.springer.com/article/10.1007/s44443-025-00133-z

Lu K, Cao Y, Chen X, et al (2021) Efficient dialog policy learning with hindsight, user modeling, and adaptation. IEEE Transactions on Cognitive and Developmental Systems 15:395–408. URL https://doi.org/10.1109/TCDS.2021.3061121

Ng AY, Harada D, Russell S (1999) Policy invariance under reward transformations: Theory and application to reward shaping. In: Proceedings of the Sixteenth International Conference on Machine Learning. Morgan Kaufmann, pp 278–287, URL https://dl.acm.org/doi/10.5555/645528.657613

Ni J, Young T, Pandelea V, et al (2023) Recent advances in deep learning based dialogue systems: A systematic survey. Artificial Intelligence Review 56:3055–3155. URL https://doi.org/10.48550/arXiv.2105.04387

Peyton K, Unnikrishnan S (2023) A comparison of chatbot platforms with the state-of-the-art sentence bert for answering online student faqs. Results in Engineering 17:100856. URL https://doi.org/10.1016/j.rineng.2022.100856

Quadros A, Silva C, Alves R (2025) Llm-driven intrinsic motivation for sparse reward reinforcement learning. arXiv preprint arXiv:250818420 URL https://doi.org/10.48550/arXiv.2508.18420

Ramachandran GS, Hashimoto K, Xiong C (2022) [caspi] causal-aware safe policy improvement for task-oriented dialogue. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 92–102, URL https://doi.org/10.48550/arXiv.2103.06370

Reimers N, Gurevych I (2019) Sentence-bert: Sentence embeddings using siamese bert-networks. 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), pp 3982–3992, URL https://doi.org/10.18653/v1/D19-1410

Saffari S, Dorrigiv M, Yaghmaee F (2025) Graph representation based reward shaping approach for addressing reward sparsity in task-oriented dialogue systems. Neurocomputing p 131058. URL https://doi.org/10.1016/j.neucom.2025.131058

Sang J, Ahmad Khan Z, Yin H, et al (2023a) Reward shaping using directed graph convolution neural networks for reinforcement learning and games. Frontiers in Physics 11:1310467. URL https://doi.org/10.3389/fphy.2023.1310467

Sang J, Wang Y, Ding W, et al (2023b) Reward shaping with hierarchical graph topology. Pattern Recognition 143:109746. URL https://doi.org/10.1016/j.patcog.2023.10974

Siddique AB, Maqbool MH, Taywade K, et al (2022) Personalizing task-oriented dialog systems via zero-shot generalizable reward function. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, pp 1787–1797, URL https://doi.org/10.1145/3511808.3557417

Snell C, Yang M, Fu J, et al (2022) Context-aware language modeling for goal-oriented dialogue systems. arXiv preprint arXiv:220410198 URL https://doi.org/10.48550/

arXiv.2204.10198

Su PH, Vandyke D, Gasic M, et al (2015) Reward shaping with recurrent neural networks for speeding up on-line policy learning in spoken dialogue systems. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp 417–421, URL https://doi.org/10.48550/arXiv.1508.03391

Sun J, Kou J, Hou W, et al (2025) A multi-agent curiosity reward model for task-oriented dialogue systems. Pattern Recognition 157:110884. URL https://doi.org/10.1016/j.patcog.2024.110884

Takanobu R, Zhu H, Huang M (2019) Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. arXiv preprint arXiv:190810719 URL https://doi.org/10.18653/v1/D19-1010

Tiwary N, Mohd Noah SA, Fauzi F (2025) Prioritising explainable AI-driven recommendations with knowledge graphs and reinforcement learning. Journal of King Saud University - Computer and Information Sciences 37:156. URL https://link.springer.com/article/10.1007/s44443-025-00173-5

Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. In: Advances in Neural Information Processing Systems, pp 5998–6008, URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Velickovic P, Cucurull G, Casanova A, et al (2017) Graph attention networks. Stat 1050:10–48550. URL https://doi.org/10.48550/arXiv.1710.10903

Wang C, Liu Y, Wang J, et al (2026a) Deep reinforcement learning for SDN routing optimization enhanced by graph multi-head attention. Journal of King Saud University - Computer and Information Sciences 38:404. URL https://link.springer.com/article/10.1007/s44443-026-00809-0

Wang H, Peng B, Wong KF (2020a) Learning efficient dialogue policy from demonstrations through shaping. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp 6355–6365, URL https://doi.org/10.18653/v1/2020.acl-main.566

Wang W, Wei F, Dong L, et al (2020b) Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. Advances in Neural Information Processing Systems 33:5776–5788. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Wang X, Wang H, Peng M, et al (2026b) Robust and interpretable fault diagnosis for nuclear power plants using fault causal graph neural networks. Engineering Applications of Artificial Intelligence 181:115569. URL https://doi.org/10.1016/j.engappai.2026.115569

Wu Y, Li X, Liu J, et al (2019) Switch-based active deep dyna-q: Efficient adaptive planning for task-completion dialogue policy learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 7289–7296, URL https://doi.org/10.1609/aaai.v33i01.33017289

Xie T, Zhao S, Wu CH, et al (2024) Text2reward: Reward shaping with language models for reinforcement learning. In: International Conference on Learning Representations, pp 35663–35699, URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/9941833e8327910ef25daeb9005e4748-Abstract-Conference.html

Xu K, Wang Z, Long Y, et al (2024) Deep reinforcement learning-based dialogue policy with graph convolutional q-network. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pp 4555–4565, URL https://aclanthology.org/2024.lrec-main.407/

Xu K, Wang Z, Zhao Y, et al (2025) An efficient dialogue policy agent with model-based causal reinforcement learning. In: Proceedings of the 31st International Conference on Computational Linguistics, pp 7331–7343, URL https://doi.org/10.1016/j.engappai.2026.115584

Xu L, Zhou Q, Gong K, et al (2019) End-to-end knowledge-routed relational dialogue system for automatic diagnosis. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 7346–7353, URL https://doi.org/10.1609/aaai.v33i01.33017346

Zhang Y, Liu N (2026) Fusing semantics and graph neural networks: A multi-dimensional survey and technological evolution. Engineering Applications of Artificial Intelligence 181:115584. URL https://doi.org/10.1016/j.engappai.2026.115584

Zhao Y, Wang Z, Yin K, et al (2020) Dynamic reward-based dueling deep dyna-q: Robust policy learning in noisy environments. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 9676–9684, URL https://doi.org/10.1609/aaai.v34i05.6516

Zhu Z, Lin K, Jain AK, et al (2023) Transfer learning in deep reinforcement learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence URL https://doi.org/10.1109/TPAMI.2023.3292075