Learning What Matters: Dynamic Experience Prioritization for Task-Oriented Dialogue Policy via Stage-aware Experience Management

Anonymous ACL submission

Abstract

Experience replay is essential for improving the sample efficiency of reinforcement learning-based dialogue policy optimization. However, traditional random sampling or static heuristic strategies fail to dynamically identify and exploit stage-critical experiences following policy learning stages, resulting in inefficient sampling and noise propagation. To address this issue, this paper proposes a novel Stage-aware Experience Management (SEM) framework that adaptively adjusts replay priorities by establishing a quantitative mapping between policy learning stages and experience states. Specifically, SEM introduces a quadripartite experience paradigm that classifies experiences into stable, forgotten, unmastered, and noisy categories, guided by real-time monitoring of loss deviations and trends with a dual Q-network. This paradigm captures the evolving utility of experiences and provides a quantitative decision-making basis for experience management. Leveraging this stage-aware mechanism, SEM emphasizes forgotten and unmastered experiences to reinforce weak decision points while suppressing noisy samples to mitigate interference. Experiments on four public dialogue datasets demonstrate that SEM achieves superior performance and strong generalizability in dynamic priority management[1].

1 Introduction

As the decision core of task-oriented dialogue (TOD) systems, dialogue policies (DPs) aim to accurately infer user intents and efficiently accomplish domain-specific goals through multi-turn interactions (Algherairy and Ahmed, 2025). Though large language models (LLMs) excel in linguistic tasks, their lack of explicit long-term value estimation undermines convergence in long-term optimization (Yi et al., 2024). Even with reasoning-enhanced prompting, LLMs still struggle to explore ambiguous solution spaces in complex dialogues efficiently, while the scarcity of domain-specific TOD datasets further exacerbates their adaptability challenges (Kamuni et al., 2024). Consequently, off-policy reinforcement learning (RL) which leverages experience replay to reuse historical interaction trajectories for policy learning has emerged as the mainstream technique for DP optimization (Algherairy and Ahmed, 2024).

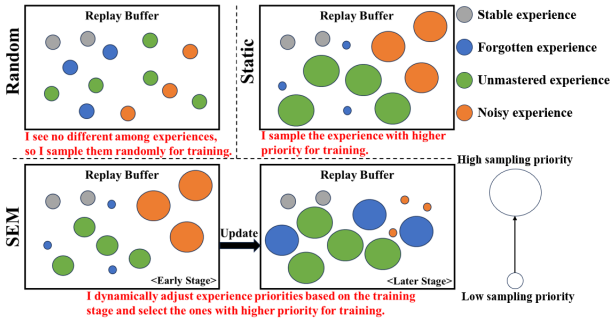


Figure 1: Comparison of SEM with random and static sampling methods, highlighting experience priority evolution from early to later stages.

The effectiveness of experience replay hinges on how sampling priorities are managed across training stages. Existing random or static heuristic strategies fail to adapt as policies evolve (Zhao et al., 2024). As shown in Fig.1, random sampling ignores experience states (Liu et al., 2024; Mnih et al., 2015), while static heuristics rely on fixed indicators that overestimate noisy experiences and prematurely discard forgotten ones, leaving them difficult to reuse later (Gu et al., 2017; Schaul et al., 2016). In short, traditional strategies cannot exploit stage-critical experiences, leading to premature dilution of high-value samples and persistent noise interference. The root cause is the absence of dynamic quantification that links policy learning phases to the true utility of experiences.

To this end, we propose Stage-aware Experience Management (SEM), a dynamic experience

[1]We will release the code on GitHub after the anonymity period.

prioritization framework designed to capture the latent interplay between experience significance and policy evolution, thereby enhancing dialogue policy optimization. Specifically, SEM dynamically assigns experiences into four states according to their alignment with the current stage of policy learning: *Stable* (consistently mastered), *Forgotten* (previously mastered but recently degraded), *Unmastered* (not effectively learned), and *Noisy* (unreliable or misleading information)[2]. These states are identified through a dual Q-network architecture. The main Q-network, updated directly with the latest training data, produces losses that reflect the policy's immediate mastery of experiences. In contrast, the target Q-network, updated periodically to avoid excessive fluctuations, provides loss signals that capture historical learning achievements. The real-time deviation and evolving trend between these two losses reveal the latent state of each experience and further guide stage-aware priority adjustment: the replay priority of forgotten and unmastered experiences is increased to remedy critical knowledge gaps in the policy, while the priority of noisy experiences is suppressed to mitigate interference during learning. To handle the computational overhead of large-scale experience storage, SEM further employs a hierarchical sum-tree structure, that supports efficient localized updates, rapid retrieval of high-priority samples, and low-complexity real-time adjustments.

In summary, the main contributions of this work are threefold:

- A quadripartite experience state quantification paradigm for TOD policy optimization is proposed, which establishes the first dynamic mapping between policy learning stages and experience values to provide a quantitative basis for experience management.
- A stage-aware experience management framework is presented, enabling real-time experience state classification and dynamic priority adjustment, with lightweight plug-and-play adaptability to off-policy RL algorithms.
- The effectiveness and generalizability of the proposed SEM are validated across four public dialogue datasets, demonstrating its capabilities in experience state identification and adaptive sampling priority management.

[2]See Sec.3.1 for their formal definition and App.A for further details.

2 Related Work

Our work focuses on improving experience management in off-policy RL-based DP optimization. Existing relevant studiesmainly rely on two paradigms: uniform random sampling and static heuristic strategies (Zhang et al., 2024).

Uniform random sampling: The foundational Deep Q-Networks (DQN) (Mnih et al., 2015) employed experience replay with uniform sampling. It treats all transitions equally regardless of their intrinsic value. Though unbiased, this method cannot distinguish valuable experiences from noisy ones, resulting in inefficient learning (Liu et al., 2024). Subsequent studies confirm its insufficiency for DP optimization where strategic experience selection proves crucial (Yang et al., 2024; Su et al., 2017; Young et al., 2007).

Static heuristic strategies: Efforts to overcome random sampling limitations have converged on three static heuristic categories: 1) Priority Weighting: Prioritized Experience Replay (PER) (Schaul et al., 2016) utilized temporal difference (TD) errors as fixed importance metrics, while variants (Mei et al., 2023; Oh et al., 2022) incorporated auxiliary reward signals. Though improving initial sampling efficiency through priority-based selection, it fundamentally operates within a binary state paradigm – identifying only mastered and unmastered experiences. 2) Noise Filtering: Approaches such as conversational dead-end detection (Zhao et al., 2024) and adversarial filtering (Yu et al., 2024) aimed to suppress low-quality samples. However, their binary keep/discard decisions often exclude borderline experiences with partial utility, exacerbating forgetting (Gu et al., 2017); 3) Memory Augmentation: Topological Experience Replay (TER) (Hong et al., 2022) organized experiences via graph structures, while multi-buffer strategies (Lu et al., 2023) preserved historical samples. Offering organizational benefits, these methods incurred prohibitive computational overhead and struggled with dynamic policy adaptation.

Despite these efforts, existing strategies commonly suffer from dual priority distortion: static priority metrics (e.g., TD-error in PER (Horgan et al., 2018)) quickly become outdated as the policy evolves, leading to premature discarding of useful experiences, while rigid noise-filtering thresholds tend to propagate residual artifacts across training iterations (Vezhnevets et al., 2017). Hybrid approaches that combine priority weighting and filter-

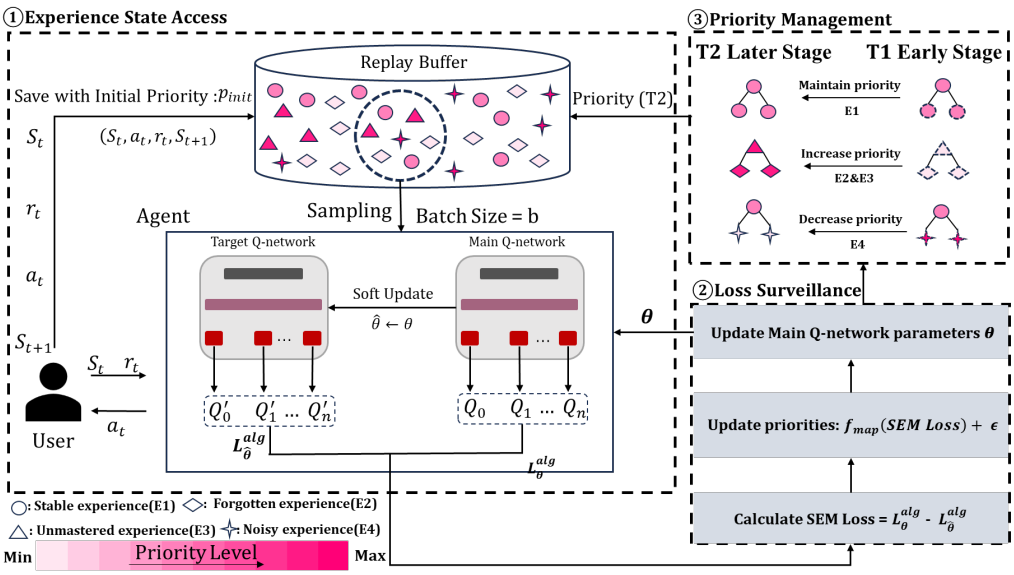


Figure 2: Experience management process for dialogue policy optimization under the SEM framework.

ing (Buzzega et al., 2020) partially mitigate these problems but still rely on heuristic assumptions and fail to support dynamic reassessment of experience utility. In contrast, our SEM framework introduces a dynamic priority calibration mechanism based on dual Q-network discrepancy analysis. This design offers three key improvements: 1) **Adaptive Reassessment**: continuous priority updates aligned with policy evolution; 2) **Noise-Resilient Selection**: probabilistic suppression rather than binary filtering; and 3) **Computational Efficiency**: logarithmic-time complexity for priority updates. By dynamically aligning experience prioritization with policy progress, SEM overcomes the fundamental limitations of previous methods while remaining fully compatible with standard off-policy RL pipelines.

3 Methodology

As shown in Fig.2, the SEM consists of three steps: 1) *Experience States Access*, which captures agent-environment interactions, systematically storing transition tuples in replay buffers partitioned into four state categories; 2) *Loss Surveillance*, which identifies four experience states by calculating loss discrepancies between the main and the target Q-networks; 3) *Priority Management*, which employs dynamic experience priority adjustment via a sum-tree architecture, enabling hierarchical storage and weighted sampling based on four experience states.

3.1 Experience State Access

The SEM framework establishes a quadripartite experience state quantification paradigm to dynamically align sampling priorities with policy learning demands. In this framework, a dual Q-network architecture is introduced, where the main Q-network loss indicates current mastery, and the target Q-network loss reflects historical learning achievements. Experiences exhibit distinct temporal loss patterns across two Q-networks, reflecting varying levels of policy mastery.

- **Stable Experiences (E1 in Fig.3)** exhibit *consistently low losses* in both main and target Q-networks, indicating mastered knowledge. These experiences should maintain baseline sampling priority to preserve policy stability.

- **Forgotten Experiences (E2 in Fig.3)** show *high main Q-network loss* paired with *low target Q-network loss*, revealing knowledge degradation. Their rising main loss triggers priority elevation to reinforce fading skills.

- **Unmastered Experiences (E3 in Fig.3)** display *Instantaneous high losses* in main Q-network, signaling temporarily unlearned patterns. These receive progressive priority boosts to accelerate initial acquisition.

- **Noisy Experiences (E4 in Fig.3)** manifest *persistent high losses* regardless of training progress. Their priority undergoes decay to mitigate interference.

This taxonomy enables the SEM adaptive experience management. During agent-environment in-

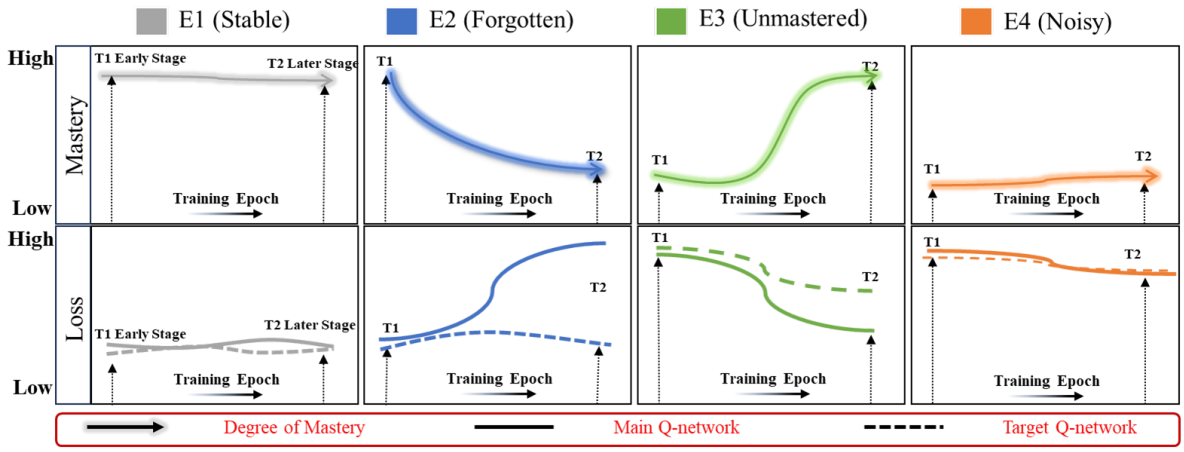


Figure 3: Evolution of main and target Q-network losses and mastery across four experience states.

teraction, the DQN agent[3] generates experience tuples $\langle \boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{r}_t, \boldsymbol{s}_{t+1} \rangle$ through ϵ-greedy exploration. The dual Q-network architecture, comprising a rapidly updated main Q-network $\boldsymbol{\theta}$ and a slowly evolving target Q-network $\hat{\boldsymbol{\theta}}$, provides temporal loss signals for state classification:

$$\hat{\theta} \leftarrow \tau\theta + (1-\tau)\hat{\theta} \tag{1}$$

$$L_{\theta}^{\text{alg}}, L_{\hat{\theta}}^{\text{alg}} = \text{TD-loss}(Q_{\theta}, Q_{\hat{\theta}}) \tag{2}$$

where $\boldsymbol{\tau} \in (\mathbf{0}, \mathbf{1}]$ is the soft update coefficient. The losses $\boldsymbol{L}_{\boldsymbol{\theta}}^{\boldsymbol{alg}}$ and $\boldsymbol{L}_{\hat{\boldsymbol{\theta}}}^{\boldsymbol{alg}}$ are computed for the SEM loss and priority update (Eq.3).

3.2 Loss Surveillance

The loss surveillance phase monitors the SEM loss of each sampled experience. It classifies the experience and updates the priority accordingly. This process consists of two steps:

1) SEM Loss: The SEM loss is the metric to measure the priority of experience. It is defined by the loss discrepancy between the main and the target Q-networks, formulated as:

$$\text{SEM Loss}_i = L_{\theta}^{\text{alg}}(i) - L_{\hat{\theta}}^{\text{alg}}(i) \tag{3}$$

where $\boldsymbol{L}_{\boldsymbol{\theta}}^{\boldsymbol{alg}}(i)$ and $\boldsymbol{L}_{\hat{\boldsymbol{\theta}}}^{\boldsymbol{alg}}(i)$ are the Q-network losses for the experience $\boldsymbol{i}$ in the main and the target Q-networks, respectively. The SEM loss provides an objective metric for prioritizing experiences by dynamically assessing agent mastery via loss discrepancies between two Q-network[4]. This design naturally supports stage-aware prioritization during policy learning.

2) Priority Update: Based on the SEM loss, we update the experience priority employing a mapping function $\boldsymbol{f_{map}}$. The form is as follows:

$$p_i = f_{\text{map}}(\text{SEM Loss}_i) + \epsilon \tag{4}$$

where $\boldsymbol{f_{map}}$ maps the SEM loss to non-negative values and adds a small positive value ϵ to ensure all experiences have positive sampling probabilities, preventing sampling dead zones (Lee et al., 2019).

After the priority is updated, the main Q-network parameters $\boldsymbol{\theta}$ will be updated.

3.3 Priority Management

The priority management phase manages the experience priorities through a hierarchical sum-tree storage structure[5].

The sum-tree is a binary tree data structure designed to manage experience priorities. Its characteristic is that each node stores the sum of its children's priorities, which allows weighted sampling and priority updates in $\mathcal{O}(\log n)$ time[6]. Moreover, when experience priority changes, only the path from the corresponding leaf to the root is updated, avoiding full array recalculations and improving update efficiency: **1) Leaf nodes:** store the priority $\boldsymbol{p}_i$ of each experience. **2) Intermediate nodes:** store the sum of the priorities of all their child nodes. **3) Root node:** stores the sum of leaf node

[3]We adopt DQN as illustrative baseline, given its seminal role in deep RL and its widespread use in task-oriented dialogue policy research. This choice ensures clear exposition and empirical validation against a well-understood off-policy RL framework. Importantly, our approach is algorithm-agnostic and can be extended to other off-policy RL–based methods, as detailed in App.I and Sec.4.4.

[4]It is worth noting that the priority calculation in PER differs from our SEM loss. PER calculates priority based on the TD errors, defined as $p_i = |\delta_i| + \epsilon$, where δ_i is the TD errors and ϵ is a small constant added to prevent zero priority. In contrast, our method adopts a different approach.

[5]See App.G and Fig.9 for more details of Sum-tree.

[6]We provide comparisons of the time and space complexity of all baselines and SEM in App.C.

Algorithm 1: SEM Implementation

Require:

- Off-policy RL algorithm $\mathcal{A}$ with loss function L^{alg}
- Main Q-network parameters $\boldsymbol{\theta}$
- Target Q-network parameters $\hat{\boldsymbol{\theta}}$
- Experience replay pool B
- Initial priority P_{init}
- SEM Loss Normalization Function f_{map}

1: Initialize Experience replay pool B as empty;

2: Initialize Target Q-network $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}$;

3: for $t = 1$ ***to*** T **do**

a. Interact with environment:;

- Observe state $\boldsymbol{s}_t$ from the environment;
- Compute action $\boldsymbol{a}_t$ from the agent;
- Execute action $\boldsymbol{a}_t$, observe reward $\boldsymbol{r}_t$, and next state $\boldsymbol{s}_{t+1}$;
- Store transition $\langle \boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{r}_t, \boldsymbol{s}_{t+1} \rangle$ in B with P_{init};

b. for *each Iteration step from 1 to* T_{iter} **do**

- Sample minibatch of size b from B;
- Compute $L^{\text{alg}}_{\boldsymbol{\theta}}$ and update $\boldsymbol{\theta}$;
- Compute $L^{\text{alg}}_{\hat{\boldsymbol{\theta}}}$ and calculate the SEM Loss;
- Update minibatch priorities: $f_{\text{map}}(\text{SEM Loss}) + \epsilon$;

4: Update Target Q-network following $\mathcal{A}$;

priorities, representing the total priority of replay buffer $\sum_{i=1}^{n} \boldsymbol{p}_i$. With this structure, a weighted sampling mechanism can efficiently sample experiences from replay buffer, balancing the latent rhythms between experience significance and policy evolution. The operation is as follows: A random value $\boldsymbol{u}$ is generated in the range $[0, \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{p}_i]$. Starting from the root node, we recursively compare $\boldsymbol{u}$ with the priorities of the left and right child nodes and eventually locate the matching leaf node (the corresponding experience tuple). It ensures that each experience is sampled with a probability $\boldsymbol{P}(\boldsymbol{i})$ proportional to its priority $\boldsymbol{p}_i$:

$$\boldsymbol{P}(\boldsymbol{i}) = \frac{\boldsymbol{p}_i}{\sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{p}_j} \tag{5}$$

The updated priorities are stored in the leaf nodes of the Sum-tree, and the priority sums of the intermediate and root nodes are recursively updated to maintain the consistency of the entire structure.

The procedure of the SEM is described in Alg.1.

4 Experiments

The objectives of this experiment are: 1) Assess the effectiveness of the SEM over baseline methods in simulated (4.2) and human evaluations (4.5); 2) Analyze the ability of SEM in handling the four experience states by the visualization of experience state distributions (4.3.1) and priority trends (4.3.2); 3) Validating the generality of the SEM in off-policy RL-based DP algorithms (4.4).

We evaluate the effectiveness of the SEM framework against six state-of-the-art experience-management baselines: DQN with Random Experience Replay (RER) (Mnih et al., 2015), which uniformly samples experiences from the replay buffer without prioritization; PER (Schaul et al., 2016), which prioritizes experiences based on high TD errors to enhance learning efficiency; TER (Hong et al., 2022), which organizes experiences into a state-dependency graph and performs value backups via breadth-first search from terminal nodes; DDR (Zhao et al., 2024), which identifies dialogue dead-ends, provides corrective rescue actions, and augments the buffer with penalty experiences to steer exploration; LLM_DA (Yi et al., 2024), which employs an LLM[7] to replace the TOD system's DP module, generating dialogue actions that are subsequently passed to an NLG component for response generation; LLM_Word (Yi et al., 2024), which extends this by using an LLM to directly select words and produce end-to-end responses.

4.1 Experimental Settings

4.1.1 Datasets

Four public datasets widely used in TODs research, including both single-domain (Li et al., 2018) and multi-domain datasets (Budzianowski et al., 2018) are employed for evaluation. The domain and feature information of datasets are shown in App.E. For simulated evaluation, we adopt three widely used metrics: success rate, average turns, and average reward. A good dialogue policy is expected to achieve a higher success rate and reward while requiring fewer turns. For human evaluation , the experimental setup is detailed in Sec.4.5.

4.1.2 Implementation Details

For experimental fairness, all dialogue agents employ identical DQN with synchronized parameter initialization. Each agent undergoes from-scratch training with uniform experience budgets (100 dialogues for warm-start initialization, 1 dialogue per

[7]We adopt GPT-4.0 as LLM-based agents for its superior generative performance on dialogue tasks. Since GPT-4.0 is closed-source and cannot be fine-tuned, we instead drive it via a carefully engineered prompt, detailed in App.G.

Domain	Agent	Epoch=50			Epoch=250			Epoch=500		
		Success↑	**Reward↑**	**Turn↓**	**Success↑**	**Reward↑**	**Turn↓**	**Success↑**	**Reward↑**	**Turn↓**
Movie	RER	0.0193	-52.49	31.62	0.4058	-4.29	27.98	0.6001	21.73	22.56
	PER	0.0163	-52.48	30.88	0.4422	0.19	27.75	0.6382	26.67	21.87
	TER	0.0152	-54.79	35.23	0.4228	-2.12	27.71	0.6698	30.90	20.96
	DDR	0.0526	-49.77	32.67	0.4788	5.53	26.89	0.7173	37.63	20.11
	LLM_DA	**0.4140**	**-4.03**	**24.88**	0.4140	-4.03	24.88	0.4140	-4.03	24.88
	LLM_Word	0.2720	-28.58	31.35	0.2720	-28.58	31.35	0.2720	-28.58	31.35
	SEM	0.0847	-43.03	28.39	**0.5806**	**18.57**	**24.21**	**0.7708**	**44.21**	**18.57**
Rest.	RER	0.0004	-43.81	29.67	0.0549	-37.85	27.58	0.2566	-18.60	25.38
	PER	0.0002	-42.10	26.23	0.1097	-32.48	26.71	0.3784	-6.39	22.90
	TER	0.0003	-41.14	24.32	0.1784	-25.86	25.83	0.3950	-4.85	22.81
	DDR	0.0020	-42.69	26.26	0.1967	-24.58	26.33	0.4601	1.80	22.26
	LLM_DA	**0.3080**	**-6.41**	**21.33**	**0.3080**	**-6.41**	**21.33**	0.3080	-6.41	21.33
	LLM_Word	0.2530	-20.31	31.63	0.2530	-20.31	31.63	0.2530	-20.31	31.63
	SEM	0.0021	-42.63	27.65	0.1826	-25.60	26.07	**0.5040**	**5.78**	**21.15**
Taxi	RER	0.0446	-39.61	29.25	0.1569	-29.52	29.29	0.2411	-21.71	28.82
	PER	0.0226	-41.51	29.08	0.1356	-31.18	28.77	0.2636	-19.21	27.87
	TER	0.0421	-39.68	28.93	0.2408	-21.41	28.17	0.3304	-13.25	27.98
	DDR	0.0265	-42.11	29.65	0.1919	-26.30	28.85	0.3550	-10.50	27.51
	LLM_DA	**0.2940**	**-12.35**	**26.75**	0.2940	**-12.35**	**26.75**	0.2940	-12.35	26.75
	LLM_Word	0.2130	-19.68	29.31	0.2130	-19.68	29.31	0.2130	-19.68	29.31
	SEM	0.0434	-39.63	29.08	**0.2949**	-16.22	27.53	**0.4551**	**-1.29**	**26.50**
MultiWOZ	RER	0.0144	-45.73	31.06	0.0287	-36.76	30.47	0.0366	-30.19	30.11
	PER	0.0156	-43.69	32.14	0.0624	-35.92	31.59	0.0842	-30.13	31.05
	TER	0.0048	-46.18	33.97	0.0533	-36.71	32.18	0.0763	-29.86	31.74
	DDR	0.0212	-41.07	30.93	0.0849	-33.42	29.17	0.1171	-25.64	29.73
	LLM_DA	**0.1220**	**-20.47**	**30.80**	**0.1220**	**-20.47**	30.80	0.1220	-20.47	30.80
	LLM_Word	0.1040	-22.57	32.78	0.1040	-22.57	32.78	0.1040	-22.57	32.78
	SEM	0.0192	-42.38	31.07	0.0995	-30.43	**28.94**	**0.1584**	**-17.91**	**27.48**

Table 1: Performance comparison of agents on four datasets with 10% noise. Top performers in each column are bolded. All results are statistically significant (t-test, $p < 0.05$). Epochs denote early(50), mid(250), and post-convergence(500) stages. GPT-4.0's results reflect its post-convergence evaluation, while all other agents were trained from scratch. Thus, we focus on comparing performance after 500 epochs with LLM-based baselines.

training epoch)[8]. A multi-layer perceptron with two hidden layers of 80 neurons each is employed across all DQN-based algorithms. For details on the experimental parameters and the implementation of the f_{map} function, please refer to the App.I.

4.2 Main Results

Tab.1 benchmarks baseline performance across four datasets[9]. RER performs worst because its uniform sampling fails to prioritize high-value experiences, hurting DP learning efficiency. PER's static reliance on TD errors induces gradient bias and delayed priority updates, degrading long-term performance. TER is unstable in practice due to its sensitivity to shifting policies. Although achieving second-best performance, DDR remains constrained by its rule-based dead-end detection mechanism that lacks online adaptability. LLM_DA achieves moderate initial success thanks to its restricted action space but cannot improve further due to hallucination issues and its closed-source nature. LLM_Word yields the poorest LLM-based results due to combinatorial action space explosion amplified by unconstrained generative hallucination tendencies. In contrast, the SEM discriminates between different experience states and adaptively adjusts their priorities, achieving best performance.

4.3 Visualization Analysis

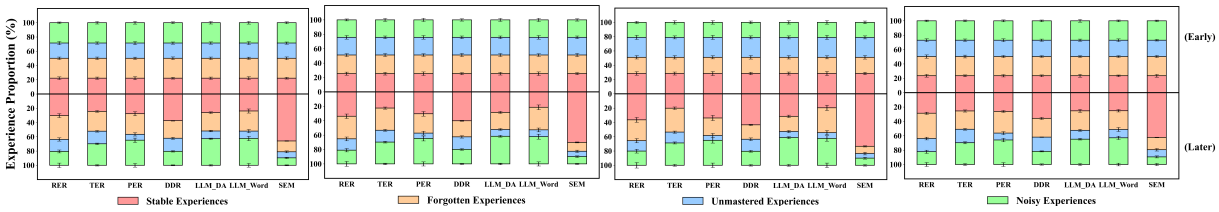
Due to space constraints, we present the experience priority trend results on the most challenging multi-domain dataset MultiWOZ 2.1 in the main text. Results on other datasets are in the App.K.

4.3.1 Evolution of Experience State Distributions

As shown in Fig.4, each color represents a different type of experience, and each subfigure corresponds to a dialogue domain, divided into two parts: the upper half depicts experience distributions during the early stage of training (epoch 50), and the lower half depicts distributions at the later stage (epoch 500). The contrast between early and late stages highlights how different strategies affect learning and experience prioritization. An effective dialogue

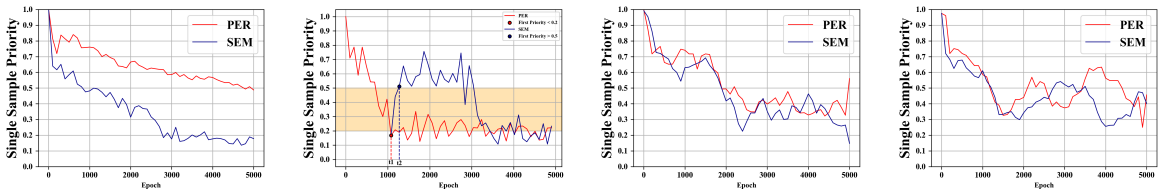
[8]A methodological gap in current dialogue policy learning research is the insufficient specification of epoch-wise data scheduling parameters. To this end, we adopt the standardized training protocol detailed by Zhao et al. (2024). While the Covlab platform reports a 0.89 success rate, it relies on a hybrid training strategy with an initial MLP pretraining (initial success rate = 0.56) followed by RL optimization. In contrast, our full retraining paradigm eliminates this pretraining advantage, under which the proposed SEM achieves significant improvements over all baselines. Moreover, for fairness, we conduct experiments with MLP pretraining followed by SEM fine-tuning, which confirm SEM's superiority (see App.D).

[9]10% noise approximates real-world dialogue system errors (8-12% range); other noise-level results appear in App.F.



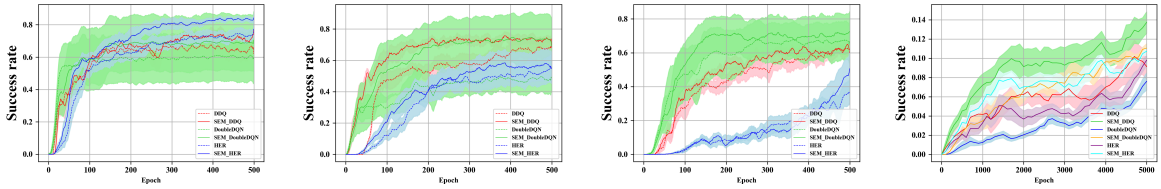
(a) Movie (b) Restaurant (c) Taxi (d) MultiWOZ 2.1

Figure 4: Evolution of experience states distribution across different baselines.



(a) Noisy (b) Forgotten (c) Stable (d) Unmastered

Figure 5: Sampling priorities for four experience states across MultiWOZ 2.1 dataset.



(a) Movie (b) Restaurant (c) Taxi (d) MultiWOZ 2.1

Figure 6: Incorporate different off-policy RL-based dialogue policy approaches with our SEM on four datasets[10].

policy is expected to achieve a higher proportion of stable experiences, with fewer forgotten, unmastered, and noisy experiences.

At the early stage, all methods exhibit similar distributions due to pre-processing that ensures a uniform initial experience allocation, making subsequent differences more apparent[11]. At later stages, RER continues to sample indiscriminately, resulting in random distributions. TER improves sampling efficiency through enhanced storage and retrieval, but also fails to differentiate experience types. PER assigns varying importance to experiences, yet cannot separate noisy from forgotten experiences, leading to simultaneous amplification of both categories. While DDR suppresses noise and forgotten states by identifying terminal-failure trajectories and re-injecting them as penalized samples, as well as through data augmentation, it is less effective on unmastered experiences. LLM-based agents introduce substantial noise due to hallucination tendencies inherent in large models. In contrast, the proposed SEM dynamically adjusts priorities for each experience type: the replay priority of forgotten and unmastered experiences is increased to remedy critical knowledge gaps, while noisy experiences are suppressed to reduce interference. This stage-aware mechanism facilitates the progressive conversion of diverse experiences into stable states during later training phases, demonstrating SEM's effectiveness in adaptive experience management across all examined domains.

4.3.2 Differentiated Efficacy Experience States

This section systematically evaluates SEM's scheduling capabilities across the four experience states: noisy, forgotten, stable, and unmastered. We specifically include PER as a baseline for three reasons: 1) As a widely-recognized sampling method, PER has proven effective in improving sample efficiency and training stability across various domains; 2) More importantly, PER employs a priority-based sampling mechanism similar to that of SEM, which establishes a common ground for fair comparison and helps to precisely isolate the source of performance differences. While both methods assign priorities to experiences, SEM introduces dynamic, stage-aware recalibration—a feature absent in PER. 3) Empirical observations indicate that SEM and PER exhibit similar priority trends for stable and unmastered experiences (Fig.5(c) and (d)), further validating PER as a suitable baseline. This similarity allows us to clearly

[10]The standalone learning curves are presented in App.I.

[11]See App.B for details on early training pre-processing.

delineate SEM's unique contributions in forgotten experience reactivation and noise suppression, which are the key differentiators of our approach.

Stable/Unmastered Experience Optimization: We visualize priority trends for stable and unmastered experiences during training. As depicted in Fig.5 (c) and (d), both methods appropriately reduce priorities for stable experiences while maintaining high priorities for unmastered ones. This alignment confirms the SEM retains PER's advantages in handling Stable/Unmastered experience states, focusing innovations on addressing PER's key limitations in noise and forgetting scenarios.

Noisy Experience Filtering: To validate the SEM's noise robustness, we inject random noise into training experiences and dynamically track their sampling priority evolution. As shown in Fig.5(a), the SEM's dual-error joint evaluation mechanism effectively identifies noisy experiences through persistent high error signals, subsequently suppressing their sampling priorities. In contrast, TD-errors-based PER fails to distinguish between meaningful gradients and noise fluctuations, persistently misidentifying noisy samples as high-priority candidates. This leads to repeated sampling of noisy experiences and degraded learning efficiency.

Forgotten Experience Reactivation: We construct a noise-free environment to isolate knowledge-forgetting, normalizing experience priorities from both methods. Thresholds ($A = 0.2, B = 0.5$) categorize experiences into *forgotten* (priority $< A$) and *reactivated* (priority $> B$). Fig.5(b) demonstrates PER's priorities drop below $A = 0.2$ at t_1, indicating gradual knowledge loss. Conversely, the SEM reactivates forgotten experiences via dual Q-network loss analysis, elevating their priorities above $B = 0.5$ at t_2. This benefit is especially pronounced in the complex taxi task and the MultiWOZ 2.1 multi-domain dialogue task.

In summary, SEM surpasses traditional priority-based methods like PER by establishing a quantitative mapping between policy evolution and experience states. Instead of PER's passive tolerance of misleading samples and rigid reliance on transient TD-error signals, SEM leverages a stage-aware mechanism to selectively reactivate forgotten experiences and suppress noisy ones, while efficiently managing stable and unmastered experiences.

4.4 Generality Evaluation

To verify the generality of the SEM, we conduct a generality evaluation by integrating it with representative off-policy RL dialogue policy methods, including DDQ (Peng et al., 2018), Double DQN (van Hasselt et al., 2016) and HER (Lu et al., 2019). As illustrated in Fig.6, the SEM exhibits method-agnostic characteristics, achieving consistent performance improvements. The superior performance metrics stem from the SEM's dual advantages in experience state differentiation and dynamic priority allocation, which complement rather than conflict with existing RL methods.

Model	Movie		Restaurant		Taxi		MultiWOZ 2.1	
	SR	HS	SR	HS	SR	HS	SR	HS
RER	0.4643	2.33	0.1786	2.55	0.2143	2.02	0.3263	2.47
PER	0.4107	2.65	0.3036	2.01	0.2321	2.14	0.3861	2.86
TER	0.5357	2.58	0.3214	2.46	0.2500	2.10	0.3790	2.82
DDR	0.4931	2.77	0.3323	2.56	0.2776	2.45	0.4028	3.30
LLM_DA	0.1968	1.68	0.1268	1.46	0.0937	1.73	–	–
LLM_Word	0.0890	1.20	0.1050	1.23	0.0366	1.09	–	–
SEM	**0.5714**	**3.05**	**0.3571**	**2.98**	**0.3036**	**2.55**	**0.4489**	**3.83**

Table 2: Human evaluation of different agents in different domains. For a fair comparison, all models in single-domain are trained for 500 epochs, while multi-domain (MultiWOZ 2.1) tests utilized RL fine-tuned models initialized with MLE-pretrained models (simulation results are detailed in App.D.)

4.5 Human Evaluation

We conducted human evaluations under the dataset platform's standardized protocol. Fifty-six evaluators (28 domain experts and 28 general users) interacted with trained models, scoring them on two metrics: Success Rate (SR, task completion) and Human Score (HS, 1-5 for naturalness/coherence). As shown in Tab.2, the SEM achieves superior performance consistent with simulation results.

5 Conclusion

This paper proposes a novel SEM framework that captures the latent rhythms between experience significance and policy evolution by quantitatively mapping dialogue experience states to policy learning states, offering new methodological insights for dynamic experience management. By analyzing loss discrepancies between the main and the target Q-networks, the SEM dynamically evaluates experience states and adjusts sampling priorities to enhance sampling efficiency. As a model-agnostic framework, the SEM can effectively integrate off-policy RL-based dialogue policy algorithms to improve performance. Extensive evaluations across four TOD datasets under various noise configurations validate the SEM's effectiveness and robustness. Visual analytics on the sampling priorities of experiences further confirms its capability to optimize priority assignments adaptively.

Limitations

The SEM demonstrates strong effectiveness in advancing prioritized experience replay, with two main limitations: First, due to its design for priority calculation, the SEM is primarily focused on optimizing experience replay in off-policy RL. In off-policy learning, historical experiences are the sole source of learning, and optimizing the priority of these experiences is crucial for improving learning efficiency and stability. However, in Online RL, real-time interactions and dynamic updates to the experience pool limit the applicability of the SEM. Future efforts will extend our method to accommodate scenarios involving Online RL. Second, the SEM dynamically adjusts the priorities of the four different experience categories, particularly emphasising the differentiated treatment of noisy and forgotten experiences. Although the SEM is an effective experience management mechanism that alleviates prioritization distortion in experience replay, it still faces the challenge of not eliminating the impact of negative experiences, as shown in Fig.4. Future research will enhance the SEM's ability to handle these situations more effectively.

Ethics Statement

We have carefully considered the potential ethical implications of our work and conclude that it does not pose significant ethical risks. Specifically:

No sensitive data involved: All experiments are conducted using publicly available or synthetic datasets that contain no sensitive personal information or private user interaction data.

Responsible human evaluation: Although human evaluation is included, it was conducted without involving identifiable individuals or personally sensitive content. No formal ethical approval was required, as the evaluation process posed no risk or harm to participants.

Low risk of misuse: The methods and data presented in this work do not enable or facilitate the generation of harmful, offensive, or otherwise unsafe content. We do not foresee any substantial risk of misuse.

Based on these considerations, our work adheres to standard ethical research practices and does not require further ethical review.

References

Atheer Algherairy and Moataz Ahmed. 2024. A review of dialogue systems: current trends and future directions. *Neural Computing and Applications*, 36(12):6325–6351.

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

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5016–5026. Association for Computational Linguistics.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, and Simone Calderara. 2020. Rethinking experience replay: a bag of tricks for continual learning. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, pages 2180–2187. IEEE.

Shixiang Gu, Timothy P. Lillicrap, Zoubin Ghahramani, Richard E. Turner, and Sergey Levine. 2017. Q-prop: Sample-efficient policy gradient with an off-policy critic. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Zhang-Wei Hong, Tao Chen, Yen-Chen Lin, Joni Pajarinen, and Pulkit Agrawal. 2022. Topological experience replay. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado van Hasselt, and David Silver. 2018. Distributed prioritized experience replay. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Navin Kamuni, Hardik Shah, Sathishkumar Chintala, Naveen Kunchakuri, and Sujatha Alla Old Dominion. 2024. Enhancing end-to-end multi-task dialogue systems: A study on intrinsic motivation reinforcement learning algorithms for improved training and adaptability. In *18th IEEE International Conference on Semantic Computing, ICSC 2024, Laguna Hills, CA, USA, February 5-7, 2024*, pages 335–340. IEEE.

Su Young Lee, Sung-Ik Choi, and Sae-Young Chung. 2019. Sample-efficient deep reinforcement learning via episodic backward update. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2110–2119.

Xiujun Li, Sarah Panda, Jingjing Liu, and Jianfeng Gao. 2018. Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *CoRR*, abs/1807.11125.

Minsong Liu, Yuanheng Zhu, Yaran Chen, and Dongbin Zhao. 2024. Enhancing reinforcement learning via transformer-based state predictive representations. *IEEE Trans. Artif. Intell.*, 5(9):4364–4375.

Cong Lu, Philip J. Ball, Yee Whye Teh, and Jack Parker-Holder. 2023. Synthetic experience replay. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Keting Lu, Shiqi Zhang, and Xiaoping Chen. 2019. Goal-oriented dialogue policy learning from failures. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 2596–2603. AAAI Press.

Yongsheng Mei, Hanhan Zhou, Tian Lan, Guru Venkataramani, and Peng Wei. 2023. MAC-PO: multi-agent experience replay via collective priority optimization. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, pages 466–475. ACM.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533.

Youngmin Oh, Jinwoo Shin, Eunho Yang, and Sung Ju Hwang. 2022. Model-augmented prioritized experience replay. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. 2018. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2182–2192. Association for Computational Linguistics.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2016. Prioritized experience replay. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Pei-Hao Su, Pawel Budzianowski, Stefan Ultes, Milica Gasic, and Steve J. Young. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, pages 147–157. Association for Computational Linguistics.

Hado van Hasselt et al. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2094–2100. AAAI Press.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. 2017. Feudal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3540–3549. PMLR.

Zhen Yang, Ming Ding, Tinglin Huang, Yukuo Cen, Junshuai Song, Bin Xu, Yuxiao Dong, and Jie Tang. 2024. Does negative sampling matter? a review with insights into its theory and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(8):5692–5711.

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

Steve J. Young, Jost Schatzmann, Karl Weilhammer, and Hui Ye. 2007. The hidden information state approach to dialog management. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2007, Honolulu, Hawaii, USA, April 15-20, 2007*, pages 149–152. IEEE.

Jiayu Yu, Jingyao Li, Shuai Lü, and Shuai Han. 2024. Mixed experience sampling for off-policy reinforcement learning. *Expert Syst. Appl.*, 251:124017.

Ye Zhang, Wang Zhao, Jingyu Wang, and Yuan Yuan. 2024. Recent progress, challenges and future prospects of applied deep reinforcement learning: A practical perspective in path planning. *Neurocomputing*, 608:128423.

Yangyang Zhao, Mehdi Dastani, Jinchuan Long, Zhenyu Wang, and Shihan Wang. 2024. Rescue conversations from dead-ends: Efficient exploration for task-oriented dialogue policy optimization. *Trans. Assoc. Comput. Linguistics*, 12:1578–1596.

A Explanation of the Four Categories of Experience Classification

We categorize experiences into four states based on the change patterns observed in the two loss functions (main and target Q-network losses) during training and their deviation from initial values. Specifically:

- **Stable Experiences** are identified when the losses of both the main and target Q-networks consistently decrease relative to their initial levels and remain stable over time. These experiences reflect mastered knowledge, where the agent has successfully learned the patterns associated with the experience. In the learning process, stable experiences show low and steady losses across both Q-networks, indicating that the agent has internalized this experience. These experiences maintain a baseline sampling priority to preserve the stability of the policy.

- **Noisy Experiences** are characterized by losses in both Q-networks that remain similar to their initial values without any significant reduction, indicating that the experience is consistently difficult for the policy to learn. Noisy experiences often contain unreliable or misleading information, such as erroneous or ambiguous states, which impedes the agent's ability to learn from them. The persistence of high losses in both the main and target Q-networks, without any significant improvement over time, signifies that these experiences do not contribute positively to policy improvement. To mitigate their negative impact, noisy experiences undergo an exponential decay in their priority, reducing their influence on training.

- **Forgotten Experiences** are recognized when the main Q-network loss first decreases and then increases during training, while the target Q-network loss remains temporarily stable (before soft updates), reflecting knowledge degradation over time. Forgotten experiences exhibit a pattern where these experiences were once learned but have since become forgotten due to changes in the policy or environment. Their rising main Q-network loss indicates that the experience is fading from memory. As a result, forgotten experiences receive an elevated priority to reinforce fading skills and help the agent recover lost knowledge.

- **Unmastered Experiences** are identified when the main and target Q-network losses initially remain undecreased, but over time, both losses decrease relative to their initial levels, indicating that the experience was not learned at first but is gradually being mastered. These experiences represent a transient state where the dialogue policy has not yet learned the experience but shows potential for learning it as training progresses. Unmastered experiences receive progressive priority boosts, which accelerate the agent's acquisition of knowledge. This gradual transition from high to low losses characterizes unmastered experiences as they move from being difficult to being successfully learned.

Therefore, instead of applying fixed thresholds, we categorize experiences based on the dynamic evolution of the losses during training, specifically how they evolve relative to their initial states.

B Details on the Experiments of Experience State Distribution Evolution

To ensure a fair comparison and clearly show how experience distribution evolves across training stages, we pre-processed and filtered the experiences to maintain a uniform distribution in the early stage. This step prevents any experience state from being overrepresented or underrepresented. Without it, unmastered experiences would dominate early training, while stable, forgotten, and noisy experiences would be secondary. Although unmastered experiences transition to stable, the reduction of forgotten and noisy experiences would be less noticeable.

(a) Early Experience State Distribution (Epoch 50): From epoch 0 to 50, each baseline begins training and gradually populates the experience replay buffer. At this stage, each agent classifies experiences by computing the SEM loss from its dual Q-networks after the target Q-network has been soft-updated. For each transition tuple (s_t, a_t, r_t, s_{t+1}), we maintain an additional label es, representing the experience state (Stable, Forgotten, Unmastered, or Noisy). To ensure a uniform distribution in the early-stage buffer, we enforce a minimum of 50 experiences per category. All other

categories are downsampled accordingly to match this minimum. This procedure ensures a balanced and fair starting point across all methods, while minimally affecting the natural categorization dynamics during subsequent training.

(b) Evolution of Later Experience State Distribution (Epoch 500): By epoch 500, each baseline has developed a relatively stable policy and a fully populated experience buffer. At this stage, we reclassify the collected experiences by analyzing the evolution of Q-network loss values over the 50 epochs surrounding epoch 500, as well as the divergence from the initial Q-values at epoch 0. This reclassification is performed using the SEM loss-based criteria to ensure consistency across different training stages. Through this process, we are able to examine how the learning dynamics of each method shape the distribution of experience states in the buffer over time.

C Time and Space Complexity Analysis

- **RER:**

  Time Complexity: The time complexity for both sampling and updating in RER is $O(1)$ because it uses a circular buffer (queue) for storage, allowing direct random access to any experience in constant time. New experiences are appended to the buffer or overwrite the oldest one.

  Space Complexity: The space complexity is $O(n)$ since it stores all experiences in a fixed-size buffer, with each experience occupying constant space.

- **PER:**

  Time Complexity: Sampling and updating in PER both take $O(\log n)$ time due to the use of a binary tree for priority management. Sampling requires traversing the tree from root to leaf, while updating priorities involves propagating changes up the tree.

  Space Complexity: The space complexity is $O(n)$ because the sum-tree requires $2n - 1$ nodes, and the experience buffer itself consumes $O(n)$ space.

- **TER:**

  Time Complexity: TER achieves linear time complexity for batch sampling ($O(B)$) by directly fetching transitions from graph vertices. Graph construction maintains $O(1)$ average-case insertion via hashing. While reverse BFS theoretically scales with graph density ($O(|V| + |E|)$), pruning enforces $|E| \leq n$ (buffer capacity), bounding updates to $O(n)$.

  Space Complexity: The graph consumes $O(|V| + |E|)$ space, where $|V|$ is the number of unique states and $|E|$ the transition count.

- **DDR:**

  Time Complexity: DDR achieves $O(n)$ time complexity for dead-end detection, where n is the buffer size. This linear scaling arises from periodic scans of the experience buffer using lightweight rule-based classifiers to identify dead-end states.

  Space Complexity: Space complexity is $O(n)$, dominated by the experience buffer storage.

- **LLM_DA / LLM_Word:**

  Time Complexity: The time complexity is dominated by LLM inference, which is $O(L^2 \cdot d_{\text{model}})$ per token for transformer-based models, where L is sequence length and d_{model} is the hidden dimension.

  Space Complexity: The space complexity is $O(P)$, where P is the number of LLM parameters (e.g., billions for GPT-3).

- **SEM:**

  Time Complexity: Sampling and updating in SEM take $O(\log n)$ time due to the sum-tree for priority management, similar to PER. The additional computation arises mainly from the losses between the dual Q-networks, which adds only minimal overhead.

  Space Complexity: The space complexity is $O(n)$ for the sum-tree and experience buffer.

D Model Performance Initialized with the Pretrained MLE Model

The results provided by the ConvLab platform are based on MLE models that have been pretrained and subsequently fine-tuned using RL, rather than models trained from scratch. Therefore, directly comparing those results to our own models trained from scratch would be unfair. To ensure a fair comparison under the same conditions, we initialized our training with the pretrained MLE model provided by ConvLab (which achieves a success rate of 0.56). We then apply RL-based baselines for fine-tuning over 2000 epochs, all uniformly based on the PPO architecture. LLM-based baselines (e.g., GPT4.0) are not available in this comparison due to their closed-source nature, which prevents us from performing any fine-tuning or reinforcement learning-based adaptation. The results, as shown in Tab. 3, further confirm that our proposed method consistently outperforms others—whether trained from scratch or fine-tuned from pretrained models.

Model	Success ↑	Rewards ↑	Turns ↓
MLE	0.56	20.9	24.1
RER	0.67	25.8	21.5
PER	0.71	35.8	22.3
TER	0.74	32.1	26.4
DDR	0.82	37.2	25.3
LLM_DA	–	–	–
LLM_Word	–	–	–
SEM	**0.89**	**44.6**	**19.0**

Table 3: Performance Comparison of Different Models Initialized with the Pretrained MLE Model

E Dataset Details

MultiWOZ 2.1 is a large-scale, multi-domain TOD dataset that includes dialogues from multiple domains such as restaurant booking, hotel booking, taxi booking, and tourist attraction recommendations. The dataset provides detailed user intent, slot annotations, and dialogue context, making it suitable for evaluating core tasks such as dialogue management, intent recognition, and slot filling. The scale and complexity of MultiWOZ 2.1 make it an ideal choice for testing cross-domain generalization ability. The Microsoft Dialogue Challenge focuses on daily conversations and customer support, offering diverse dialogue scenarios across three domains: movie-ticket booking, restaurant booking, and taxi booking, making it suitable for multi-task learning and sentiment analysis research. By training with these datasets, this study can validate the effectiveness of the proposed SEM loss method in multi-domain, multi-task environments, particularly in handling noisy experiences, forgetting experiences, and human evaluation performance. Details of the datasets are provided in Tab.4.

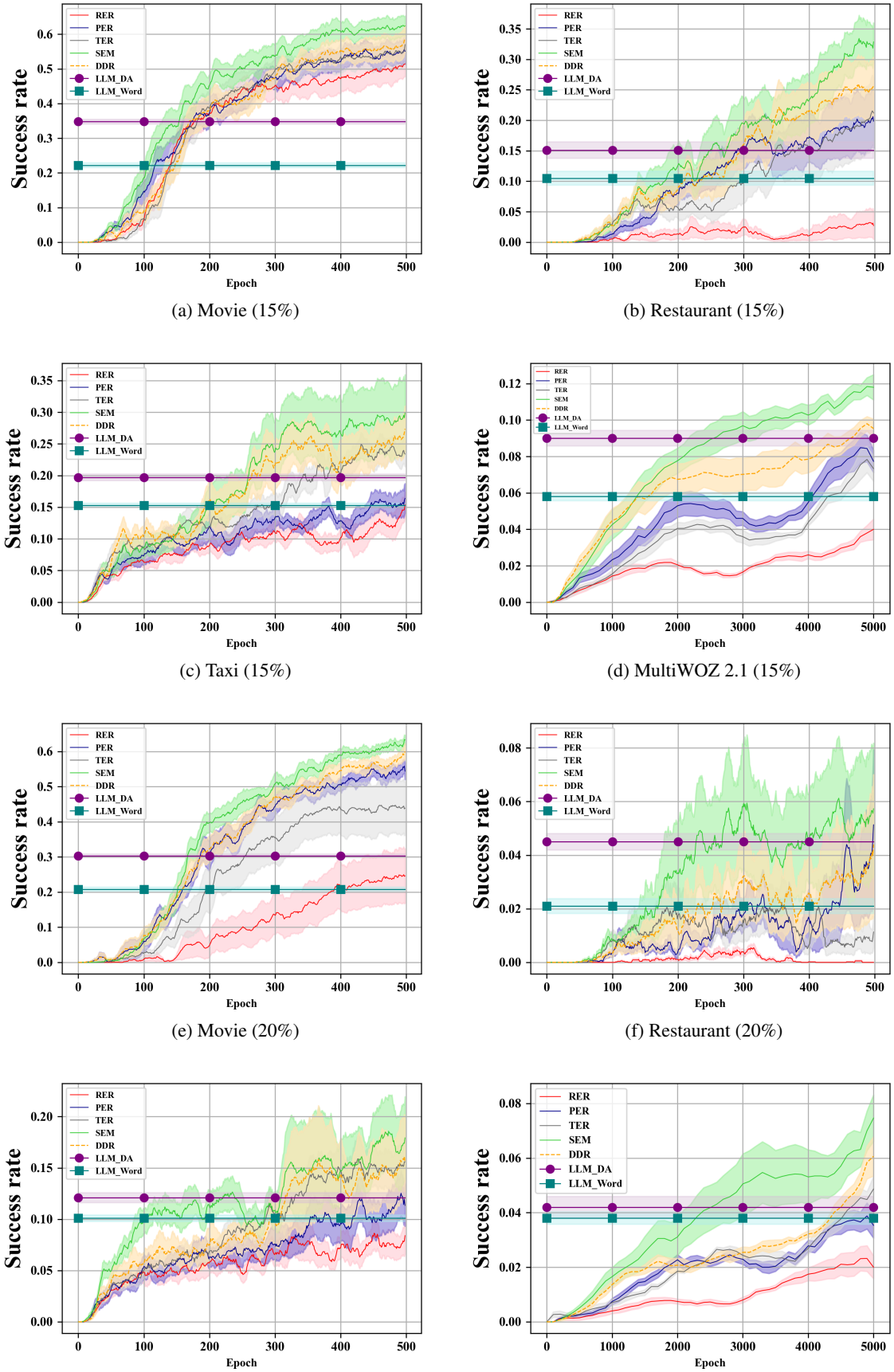
Dataset	Domains	Scale
MultiWOZ 2.1	7	Dialogue scale: 8,438; Dialogue rounds: 115,424; Avg. rounds: 13.68; Slots: 25
Movie	1	Dialogue scale: 2,890; Intention: 11; Slots: 29
Restaurant	1	Dialogue scale: 4,103; Intention: 11; Slots: 30
Taxi	1	Dialogue scale: 3,094; Intention: 11; Slots: 29

Table 4: Dataset statistics for various dialogue tasks.

F Performance Comparison under Varying Levels of Noise

As shown in Fig. 7, the SEM outperforms all other methods in all scenarios, demonstrating higher success rates and better stability, especially in high noisy environments (15%&20%). It highlights its ability to identify and avoid noisy experiences effectively. Although PER performs better than RER, its performance still lags behind the SEM. The increased probability of noisy experiences in such environments leads PER to repeatedly prioritize this experience, which is confusing training. In contrast, the SEM adjusts the priority of noisy experience by comparing losses between the main and marget Q-networks, ensuring that its priority decreases as training progresses, avoiding confusion. For TER, using hash tables to construct the graph optimization experience replay process is better than RER to a certain extent. Still, its performance will be limited when the task state and action space increase sharply and are disturbed by noisy experiences.

Furthermore, we conducted experiments under ideal conditions without the presence of noise, as depicted in Fig. 8. Even in such scenarios, our method continues to deliver superior performance. This outcome eliminates the influence of noisy experiences, highlighting the impact of other experience states, particularly forgotten experiences.



(a) Movie (15%)

(b) Restaurant (15%)

(c) Taxi (15%)

(d) MultiWOZ 2.1 (15%)

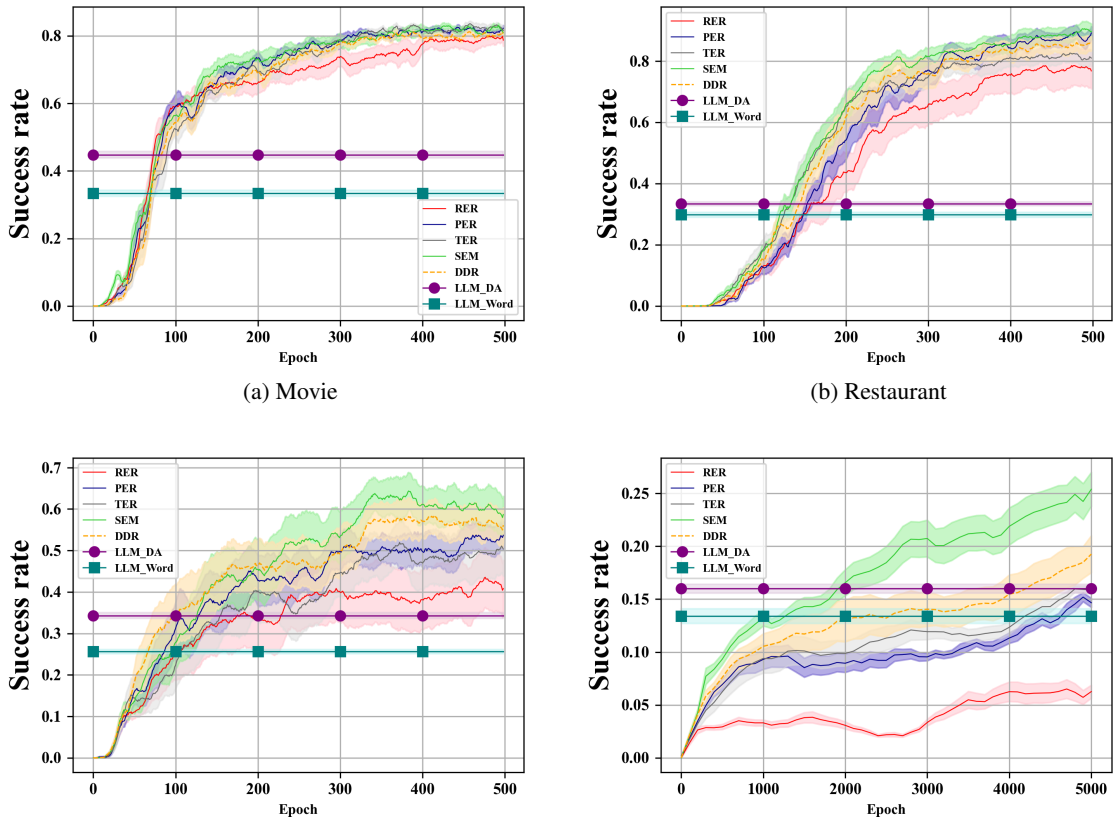
(e) Movie (20%)

(f) Restaurant (20%)

(g) Taxi (20%)

(h) MultiWOZ 2.1 (20%)

Figure 7: Performance comparison of agents under noisy environments (15% and 20% noise) across four datasets.



(a) Movie (b) Restaurant

(c) Taxi (d) MultiWOZ 2.1

Figure 8: Performance comparison of agents under noisy-free environment) across four datasets.

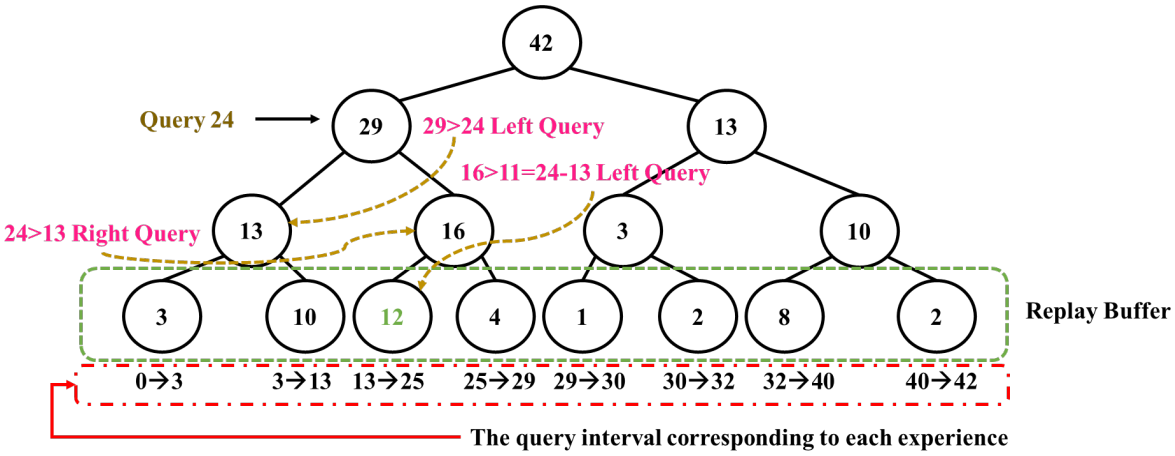


Figure 9: Sum-tree mechanism

G Sum-tree

- Node Definition:

  1. The overall structure is a binary tree, with each parent node having only two child nodes.

  2. The leaf node represents each sample's priority index -> SED Loss. Leaf nodes contain the Experience Replay Buffer information containing the priority index, from which training data is sampled.

  3. The intermediate node represents the sum of the priority indexes of the child nodes.

  4. The root node represents the sum of the priority indexes of all samples.

- The process of generating a sum-tree:

  1. Leaf nodes serve as replay buffers. Like DQN, they are updated cyclically with a fixed queue length, without sorting.

  2. Intermediate nodes and the root node are updated according to the Sum method in the figure above.

  3. Note that the brackets around each leaf node in the figure above indicate the query range corresponding to the leaf node. Nodes with higher priorities have wider query ranges and are more likely to be sampled.

- The specific sampling process is as follows (Please refer to the Fig.9):

  1. Based on the size of the learning mini-batch b, divide the sum-tree root node (the sum) into b segments, and evenly sample the query priority cumulative value in each segment.

  2. For each sampled value, execute steps 3-6 to locate the Experience to be sampled.

  3. For example, in the figure, the query value 24 is sampled.

  4. If the left node 29 is greater than the query value 24, continue traversing along the left node.

  5. If the left node 13 of 29 is less than the query value 24, traverse along the right node 16 and update the query value 11 = 24 - 13. Note that this is an accumulative tree, which is easier to understand through backtracking. During backtracking, the query value + 13 here is the query value at the parent node.

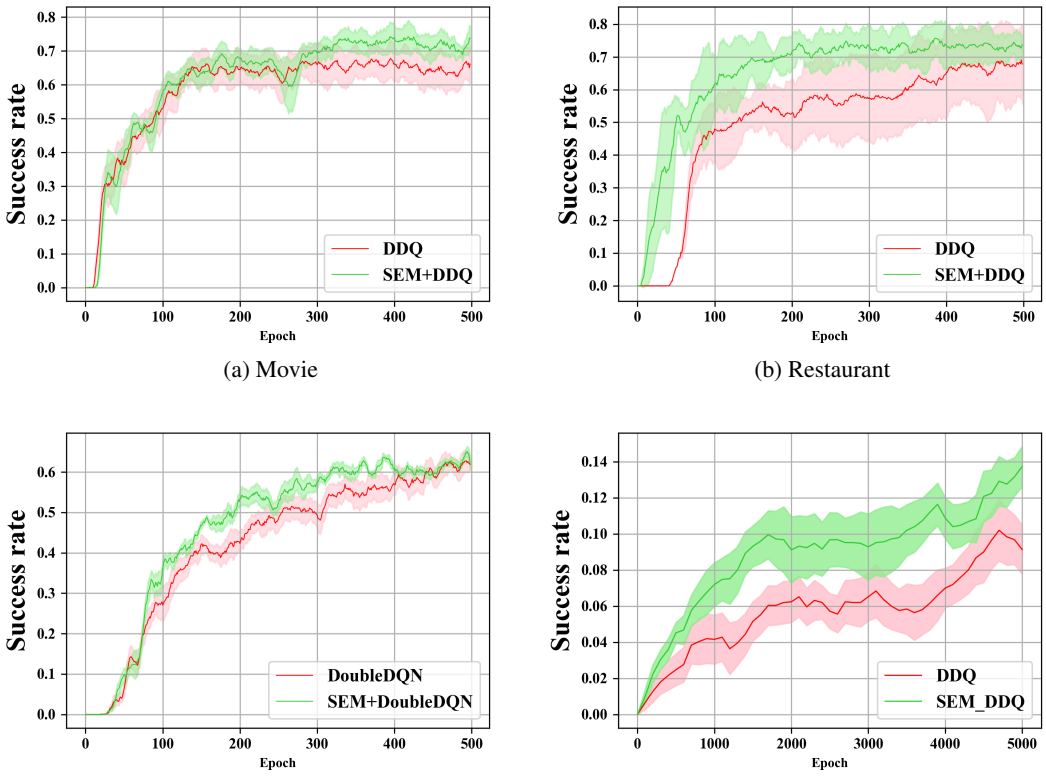
  6. If the left node 12 of 16 is greater than the query value 11, the leaf node with priority 12 is ultimately selected.

H Prompt Design

Model	Prompt
LLM_DA	1. System role definition: Function as the policy component in of a task-oriented dialogue system, you need to produce system responses according to dialogue context. 2. Processing user dialogue state: Process the provided dialogue state representation to guide response selection. 3. Generate system actions: Given the user's dialogue state, generate system actions in the format: [["ActionType", "Domain", "Slot", "Value"]]. Here, 'ActionType' refers to the system's intended operation (e.g., Request, Inform, Confirm), 'Domain' indicates the relevant area (such as restaurant, taxi, or hotel), 'Slot' specifies the information field (like name, area, or type), and 'Value' holds the related content or remains empty if not provided. 4. Compliance requirements: Output must strictly conform to JSON schema without extraneous content.
LLM_Word	1. System role definition: Function as the dialog policy module and natural language generation module of task-oriented dialogue systems, you need to determine system behaviors based on real-time conversational state analysis. 2. Processing user dialogue state: Process the provided dialogue state representation to guide response selection. 3. Generate system actions: make decisions based on the current state of the dialogue and formulate natural language responses to the user. 4. Compliance requirements: Output must strictly conform to JSON schema without extraneous content.

Table 5: Descriptions of Prompts used for LLM-based baselines.

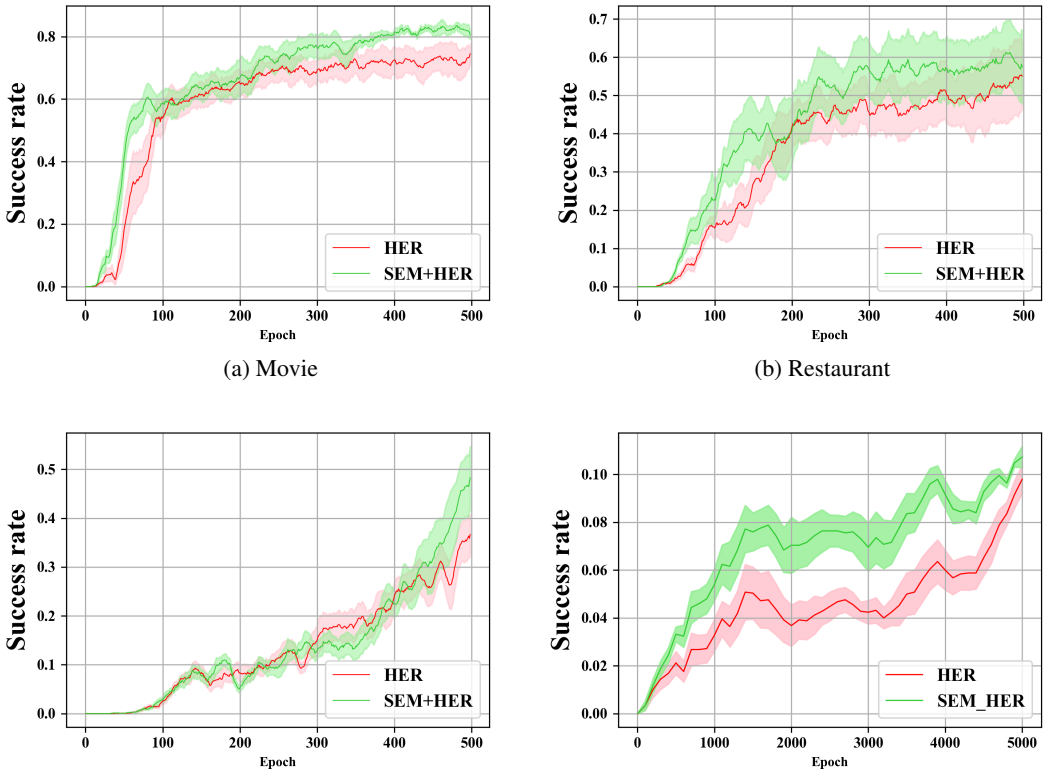
I Generality Evaluation of SEM-based Methods



(a) Movie (b) Restaurant

(c) Taxi (d) MultiWOZ 2.1

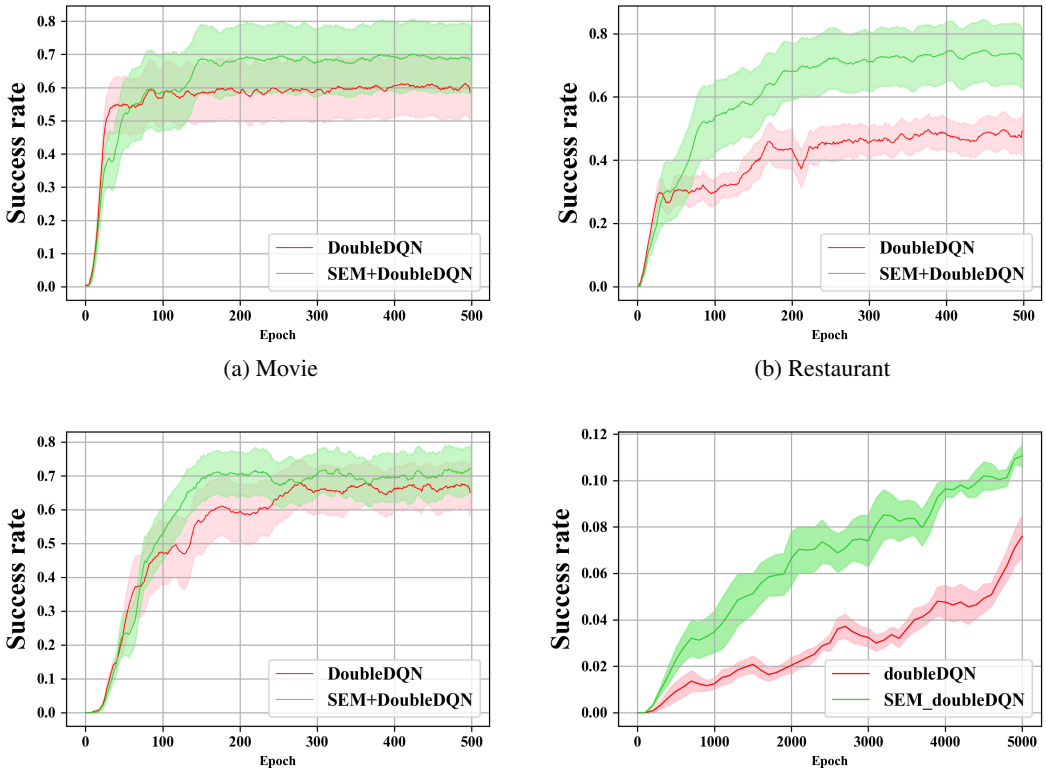
Figure 10: Incorporating DDQ with our SEM on four datasets.



(a) Movie (b) Restaurant

(c) Taxi (d) MultiWOZ 2.1

Figure 11: Incorporating HER with our SEM on four datasets.



(a) Movie (b) Restaurant

(c) Taxi (d) MultiWOZ 2.1

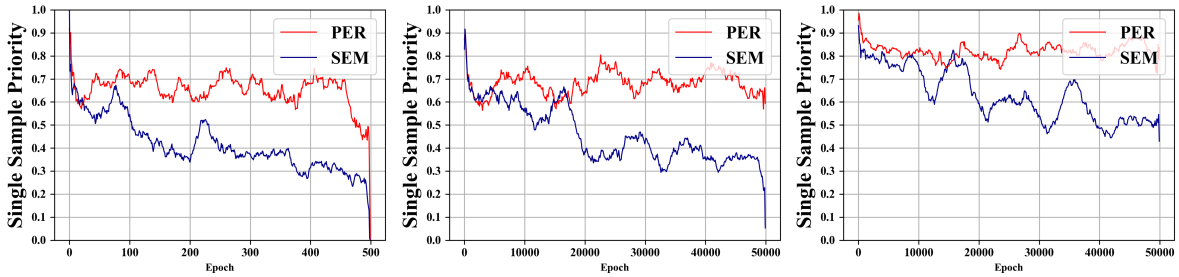
Figure 12: Incorporating Double DQN with our SEM on four datasets.

J Experimental Setup Details

Parameter	Value / Description
Exploration rate (ϵ)	Initial: 0.1; decayed to 0.01 during training
Update coefficient (τ)	1×10^{-2} (soft update for target Q-network)
L2 regularization coefficient (regc)	1×10^{-3}
Discount factor (γ)	0.99
Batch size	16
Learning rate	0.001
Replay buffer size	10,000
Operating system	Ubuntu 24.01
Python version	3.9
Toolkit	ConvLab-3
Simulator	Rule-based simulator from ConvLab-3
Model architecture	Modular neural models
NLU component	SVM-based semantic parser (SVMNLU)
NLG component	Template-based generator (TemplateNLG
State tracker	Rule-based dialogue state tracker (RuleDST)
Fmap	ReLU(x)=max(0,x)

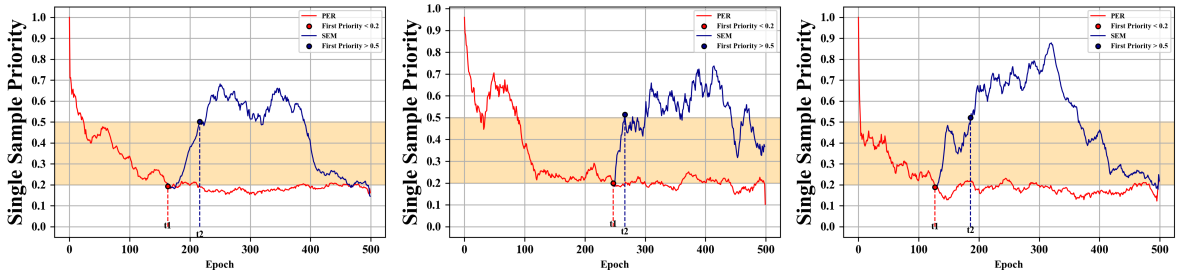
Table 6: Experimental Settings

K Sampling Priorities for Four Experience States across Movie, Restaurant, Taxi Datasets



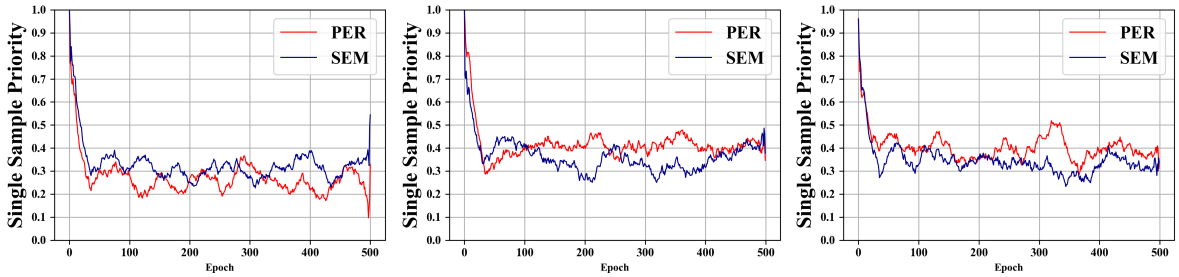
Movie Restaurant Taxi

(a) Sampling priorities for noisy experiences across three datasets.



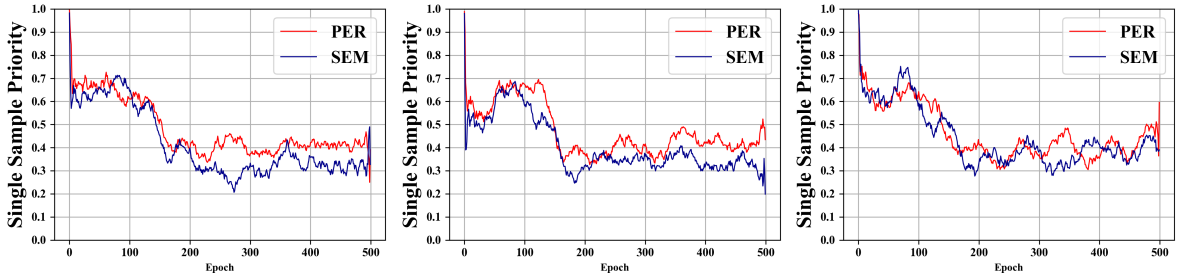
Movie Restaurant Taxi

(b) Sampling priorities for forgotten experiences across three datasets.



Movie Restaurant Taxi

(c) Sampling priorities for stable experiences across three datasets.



Movie Restaurant Taxi

(d) Sampling priorities for unmastered experiences across three datasets.

Figure 13: Comparative analysis of priority trends for four experience states in SEM and PER.