

From Monolithic Prioritization to Stratified Replay: Hierarchical Prioritized Experience Replay for Task-Oriented Dialogue Policy Learning

Yixin Zhang¹, Yangyang Zhao^{1*}, Kai Xu², Lingyun Xiang¹, and Yun Song¹

¹Changsha University of Science and Technology

²Guangdong Polytechnic Normal University

zyxin@stu.csust.edu.cn, yyz@csust.edu.cn

Abstract

Replay sampling is central to off-policy task-oriented dialogue policy learning, as it determines which past interactions are revisited during policy updates. Existing priority-based replay methods often rely on a monolithic priority distribution to form minibatches, which entangles coverage across error magnitudes with emphasis on high-error transitions. Under sparse dialogue success and heavy-tailed temporal-difference (TD) errors, this design overexposes extreme residuals and suppresses informative medium-error transitions, a failure mode we call *mid-magnitude starvation*. To address this problem, we propose **Hierarchical Prioritized Experience Replay** (HPER), a two-level replay sampler that separates coverage from prioritization. HPER first samples TD-error quantile strata according to occupancy, ensuring that different error magnitudes enter training, and then applies local prioritized replay within the selected stratum to retain emphasis on large residuals. This coverage-first design restores intermediate learning signals without diluting high-error samples. Experiments on multiple dialogue datasets show that HPER improves replay efficiency and yields faster competence onset and higher terminal success than uniform replay and monolithic priority-based replay under matched budgets.

1 Introduction

Task-oriented dialogue systems require policies that can make reliable decisions across multi-turn interactions, where each action may affect both the immediate user response and the eventual task outcome. Reinforcement learning provides a natural framework for optimizing such policies through interaction, directly targeting long-term task success rather than isolated turn-level prediction (Young

et al., 2013; Li et al., 2016; Peng et al., 2018). However, dialogue policy learning is often challenging because task success signals are sparse, rewards are delayed, and many intermediate turns provide only weak supervision. Under these conditions, how effectively an agent reuses past interactions becomes critical to sample efficiency and stable policy improvement.

Experience replay is a standard mechanism for improving experience reuse in off-policy reinforcement learning (Mnih et al., 2015). By storing past transitions and revisiting them during updates, replay allows the agent to learn repeatedly from informative interactions. Priority-based replay further biases sampling toward transitions with large temporal-difference errors, based on the intuition that larger prediction errors indicate stronger learning signals (Schaul et al., 2016). This strategy is attractive for task-oriented dialogue, where rare failures, recovery points, and high-impact decisions can provide important corrective information.

Despite its effectiveness, priority-based replay can introduce a structural imbalance in dialogue policy learning. Existing methods often sample from a single monolithic priority distribution over the entire replay buffer. As shown in Figure 1, this design entangles two different goals: covering diverse error magnitudes and emphasizing high-error transitions. When TD errors are heavy-tailed, a small number of extreme-error samples can dominate the replay distribution. As a result, minibatches may repeatedly focus on the global high-error tail while under-sampling medium-error transitions.

This under-sampling is problematic because medium-error transitions are often highly informative in dialogue learning. They may correspond to partially successful trajectories, recoverable mistakes, ambiguous states, or interactions near the boundary between failure and success. Although

*Corresponding author.

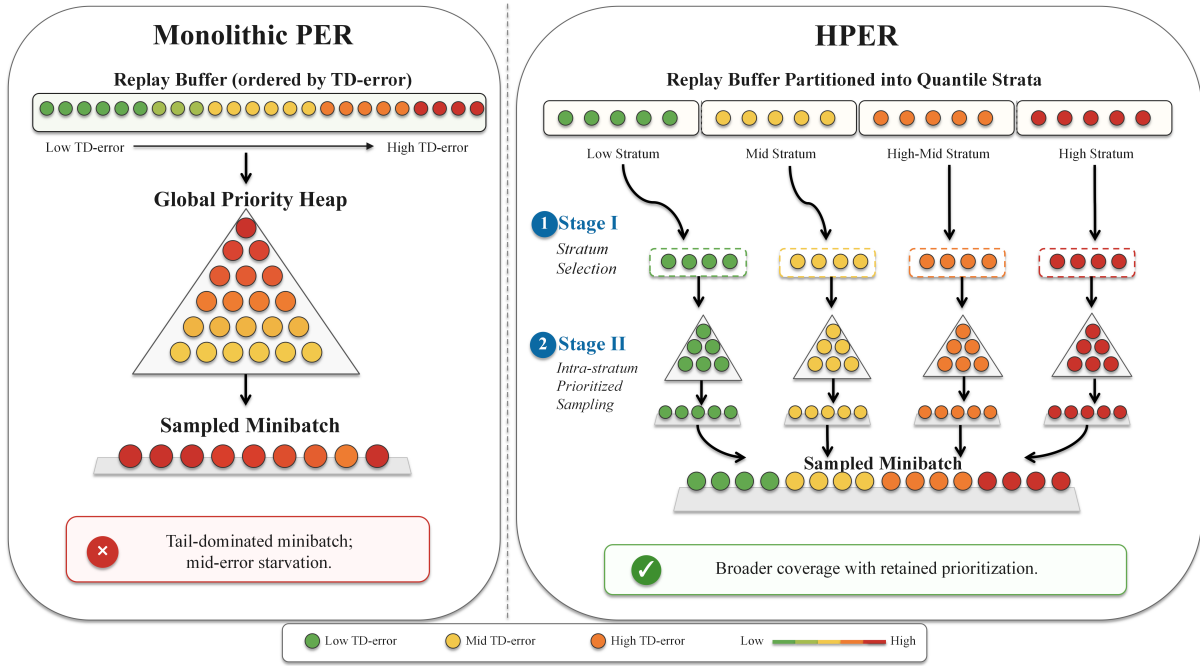


Figure 1: **Qualitative contrast of minibatch composition.** *Left:* monolithic priority replay uses one global TD ranking, so a few very large errors dominate each minibatch and medium-error transitions are seldom batched together. *Right:* HPER first draws a TD-error stratum by buffer occupancy (Stage I), then prioritizes within that stratum (Stage II), separating which error scales enter training from which transitions receive extra weight inside a scale.

such transitions may not produce the largest TD errors, they provide the intermediate corrective signals needed for gradual policy refinement. We call this failure mode **mid-magnitude starvation**: a replay-level imbalance where extreme residuals are overexposed and informative medium-error experiences are suppressed.

To address this issue, we propose **Hierarchical Prioritized Experience Replay**. The core idea is to separate coverage from prioritization. Instead of sampling directly from one global priority distribution, HPER first partitions the replay buffer into TD-error quantile strata. It then samples a stratum according to its occupancy, ensuring that different error ranges are routinely represented during training. Within the selected stratum, HPER applies local prioritized replay, preserving emphasis on larger residuals without allowing a small global tail to dominate the entire minibatch. Figure 1 illustrates this contrast qualitatively. This hierarchical design makes replay coverage-aware while retaining the benefits of prioritization. Medium-error transitions can enter training regularly, while high-error samples remain emphasized within their own error range. Importantly, HPER is lightweight: it modifies only the replay sampling process and does

not require changes to the dialogue policy architecture, learning objective, or environment.

We evaluate HPER on the single and multiple domain datasets and MultiWOZ 2.1 datasets under matched training budgets, with blind human evaluation on both settings. Results show that HPER improves replay efficiency over uniform replay and monolithic priority-based replay, yielding faster competence onset and higher terminal success. Human judgments corroborate the ranking. These findings suggest that replay efficiency depends not only on prioritizing high-error transitions, but also on maintaining balanced coverage over the broader TD-error landscape.

In summary, our contributions are threefold:

- We identify mid-magnitude starvation as a replay-level failure mode in priority-based task-oriented dialogue policy learning.
- We propose HPER, a two-level replay sampler that separates error-range coverage from intra-stratum prioritization.
- We show that HPER improves replay efficiency, yielding faster policy competence acquisition and stronger final task success under matched budgets.

2 Related Work

Task-oriented dialogue policy learning is commonly formulated as a sequential decision-making problem under partial observability, where reinforcement learning optimizes long-term task success through interaction with users or simulators (Young et al., 2013; Peng et al., 2018). Recent work improves dialogue policies through stronger exploration, reward design, or policy optimization under simulated environments (Zhao et al., 2024; Kaiser et al., 2024). These studies establish off-policy reinforcement learning as an effective framework for task-oriented dialogue, but they mainly focus on the policy learning framework, environment, or optimization objective. Our work is complementary: we keep the dialogue environment, reward, action interface, and learner fixed, and ask how replay sampling alone shapes policy learning.

Experience replay is a standard component of off-policy value-based learning, with deep Q-network (DQN) typically reusing past transitions through uniform sampling (Mnih et al., 2015; van Hasselt et al., 2016). Prioritized experience replay (PER) improves sample reuse by sampling transitions according to temporal-difference error magnitude and correcting the induced bias with importance-sampling weights (Schaul et al., 2016). Subsequent work refines priority estimation or analyzes the interaction between prioritization, bootstrapping, and function approximation (Panahi et al., 2024; Li et al., 2024; Pleiss et al., 2025). These methods generally operate over a single global priority distribution. In contrast, HPER targets the coupling in monolithic prioritization by separating coverage across TD-error magnitudes from additional emphasis within each magnitude range.

Another line of work improves replay by introducing additional structure, such as uncertainty-aware sampling, imbalance correction, diversity-aware selection, or task-aware allocation (Carrasco-Davis et al., 2025; Hong et al., 2023; Zhao et al., 2025). Hierarchical or transfer-oriented replay schedules further show that organizing experience can affect learning dynamics, but they typically route experience across tasks rather than stratifying one buffer by TD-error scale (Yin and Pan, 2017). HPER differs by keeping one replay buffer and one learner while partitioning transitions into TD-error quantile strata. This design implements cross-scale coverage and within-scale prioritiza-

tion as two separate stages of the sampler, rather than relying on a single global priority map. For curve-level summaries we follow common deep RL reporting conventions (Henderson et al., 2018; Agarwal et al., 2021), including paired bootstrap comparisons where they help interpret seed variance. We further evaluate discrete-action ports of the replay refinements by Li et al. (2024); Pleiss et al. (2025) as replay-side references under the same training stack. These ports drop continuous-control-only machinery and are not intended as faithful reproductions of the original benchmarks.

3 Background

We adopt the usual deep Q -network setup with replay and a slowly updated target network, parameterizing $Q(s, a; \theta)$ and $Q(\cdot; \theta^-)$ for stable bootstrapping. Under any replay distribution over transitions $(s_t, a_t, r_t, s_{t+1}, d_t)$, training minimizes the usual TD loss,

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s',d) \sim \mathcal{D}} \left[(y - Q(s, a; \theta))^2 \right], \quad (1)$$

$$y = r + \gamma(1 - d) \max_{a'} Q(s', a'; \theta^-), \quad (2)$$

where d is an episode-termination flag, γ is the discount factor, and $\mathcal{D}$ denotes the (possibly non-uniform) replay distribution induced by the sampling procedure. Following PER (Schaul et al., 2016), non-uniform draws require per-transition importance-sampling (IS) weights so that the stochastic minibatch objective remains a valid proxy for the intended off-policy gradient direction. HPER redefines only how $\mathcal{D}$ is defined for minibatch construction, and all other DQN components remain unchanged.

4 Methodology

As shown in Figure 2, HPER decouples *coverage across TD-error scales* from *emphasis within a scale*. The method comprises three modules: (1) a unified replay buffer with quantile TD-error strata and decoupled priority metadata, (2) a stratified minibatch sampler that routes draw mass by stratum occupancy before applying prioritized sampling within the selected stratum, and (3) the standard DQN TD update and replay buffer maintenance applied to weighted minibatches,¹ where the

¹Structured and prioritized replay is commonly developed under deep Q -learning (Mnih et al., 2015; van Hasselt et al., 2016; Schaul et al., 2016; Hessel et al., 2018; Li et al., 2024; Pleiss et al., 2025). We therefore present and evaluate HPER

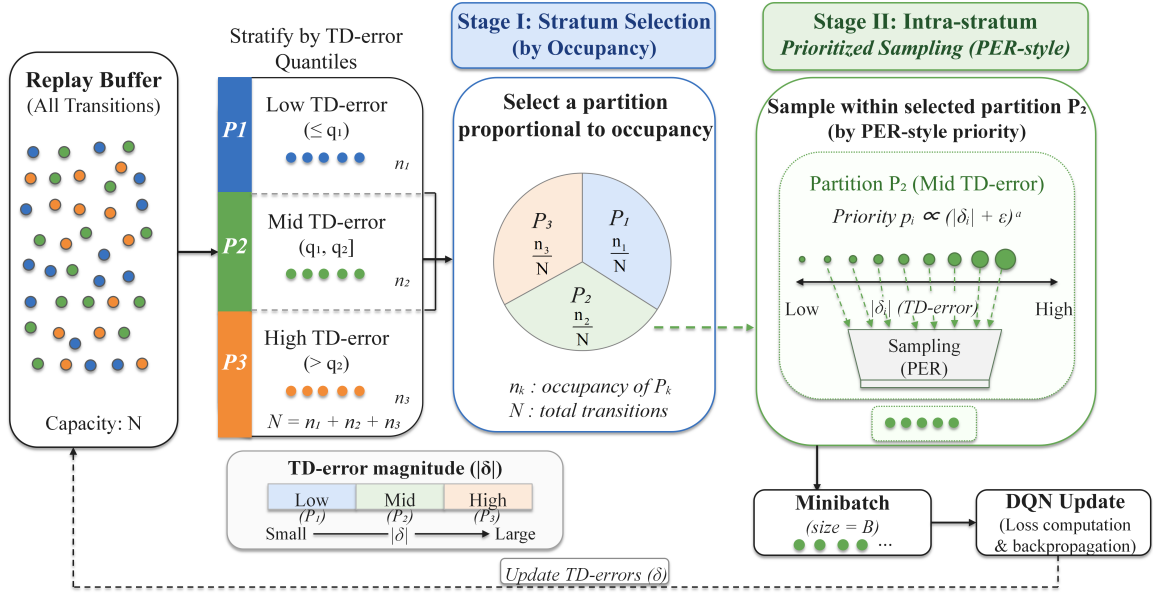


Figure 2: Overview of HPER in the DQN training loop. Transitions in one replay buffer are partitioned into quantile TD-error strata (P_1, P_2, P_3 shown), then sampled in two stages. Stage I selects a stratum proportional to occupancy, and Stage II performs prioritized intra-stratum sampling to form the minibatch. The sampled minibatch drives the standard TD update and refreshes TD-error priorities for the next round. *Special cases.* $K=1$ reduces to monolithic PER, and $\alpha=0$ keeps stratified mixing but uses uniform intra-stratum sampling.

replay distribution $\mathcal{D}$ is induced by (2) over the memory in (1). Monolithic PER entangles these roles in one global heap, whereas HPER keeps the Bellman backup fixed and treats them as separate design choices.

4.1 Replay buffer with quantile strata

A naive stratification could duplicate transitions across parallel buffers or tie stratum membership too tightly to the same scalar used for prioritized draws, blurring the separation HPER targets. We therefore keep one shared pool and annotate each slot with a stratum identifier $c_i \in \{1, \dots, K\}$ that partitions the buffer into TD-error bands $P_1, \dots, P_K$ (Figure 2). Each transition maintains (i) a *sampling priority* p_i for Stage II and (ii) a TD-error proxy δ_i used only to refresh c_i , with K as a hyperparameter (implementation flag `hpr_partitions`). On a fixed schedule, quantile thresholds are recomputed on $\{\delta_i\}$ and each c_i is assigned by binning δ_i . When quantiles are unstable (early training, small buffers, or many near-tied δ_i), we fall back to *equal-count* rank buckets (sort by δ_i ,

with DQN. The replay sampler is largely backbone-agnostic and intended to transfer to other off-policy learners in the same way.

split into K near-equal segments). This schedule isolates “which stratum a transition occupies” from “how aggressively it is resampled inside that stratum”. p_i and δ_i may be updated asynchronously in code, with defaults tabulated with other hyperparameters.

4.2 Stratified minibatch sampling

Given $\{c_i\}$, minibatch construction repeats until B transitions are collected, following the Stage I $\rightarrow$ Stage II pipeline in Figure 2.

Stage I (occupancy-based routing). Let n_k be the count of transitions with $c_i=k$ and $N = \sum_k n_k$. We draw stratum k with

$$\pi_k = \frac{n_k}{N}, \quad (3)$$

so Stage I is occupancy-weighted: each error-scale band is selected in proportion to its buffer mass. This is not uniform over strata (a fixed $1/K$ quota) and not uniform over transitions (as in DQN). A fixed $1/K$ mix would oversample thin, high-variance bands early on. Under heavy-tailed TD errors, the moderate-magnitude band typically carries the largest n_k , so Stage I repeatedly exposes it to learning even when a global PER heap would assign most probability mass to a sparse extreme

tail. Tail-heavy emphasis is deferred to Stage II and applies only after a band has been selected.

Stage II (conditional prioritized sampling). Given k , let $\mathcal{I}_k$ denote indices with $c_i=k$. We draw $i \in \mathcal{I}_k$ with conditional PER mass

$$\pi(i | k) = \frac{p_i^\alpha}{\sum_{j \in \mathcal{I}_k} p_j^\alpha}, \quad (4)$$

with exponent α (implementation `hpr_alpha`). Setting $\alpha=0$ yields uniform draws within $\mathcal{I}_k$ while keeping Stage I unchanged (a hierarchy-only ablation). The joint draw probability is $\pi(i) = \pi_k \cdot \pi(i | k)$, and IS weights follow the usual PER form,

$$w_i = \left(\frac{1}{N \pi(i)} \right)^\beta, \quad (5)$$

with exponent β (implementation `hpr_beta`, with optional annealing in code, but comparisons fix β). Finally, $K=1$ collapses strata, Stage I selects the single band, and Stage II reduces to flat PER over the full buffer, recovering the standard PER baseline.

4.3 DQN training integration

Once transition indices and IS weights are assembled, the remainder of an optimizer step matches DQN/PER. Replay variants share the Bellman backup and action interface but differ in minibatch construction, importance-sampling weights, and buffer maintenance at capacity (Table 7). For PER and HPER, buffer maintenance follows standard PER practice (eviction by lowest priority, high priority on fresh inserts, TD-driven priority refresh) (Schaul et al., 2016); DQN uses uniform FIFO replacement. The primary PER–HPER comparison therefore isolates stratified sampling rather than eviction policy; HPER additionally relabels c_i on scheduled quantile refreshes (Section 4.1). HPER adds vectorized stratum bookkeeping and periodic threshold updates but introduces no extra forward or backward passes relative to flat PER, and observed run time is still dominated by the usual DQN forward and backward cost.

Algorithm 1 condenses one training step as implemented in `agent_dqn.py`, including optional stratum relabeling, the Stage I–II loop (Eqs. (3)–(5)), PER-style weight clipping and batch normalization, one gradient step, and exponential moving averages for p_i and δ_i .

Remark. Special cases $K=1$ and $\alpha=0$ are discussed in Section 4.2. Section 5 specifies the con-

Algorithm 1: HPER minibatch sampling and priority update (condensed)

Input: Buffer with (p_i, δ_i, c_i) , batch B , quantile-refresh schedule, α , and β .

Output: One DQN step on θ and updated p_i, δ_i, c_i for touched indices.

```

1 if quantile refresh is due then
2   | Reassign all  $c_i$  by  $K$ -way stratification of  $\{\delta_i\}$ 
   | via quantiles, and if unstable, use equal-count
   | rank buckets
3  $\mathcal{M} \leftarrow \emptyset$ 
4 while  $|\mathcal{M}| < B$  do
5   |  $n_k \leftarrow |\{i : c_i=k\}|$ ,  $N \leftarrow \sum_k n_k$ , draw  $k$  with
   |  $\pi_k = n_k/N$  (Eq. (3))
6   |  $\mathcal{I}_k \leftarrow \{j : c_j=k\}$ 
7   | if  $\alpha=0$  then
8   |   | draw  $i$  uniformly from  $\mathcal{I}_k$ 
9   | else
10  |   | draw  $i \in \mathcal{I}_k$  with  $\mathbb{P}(i | k) \propto p_i^\alpha$  (Eq. (4))
11  |   |  $\pi(i) \leftarrow \pi_k \pi(i | k)$ ,  $\tilde{w}_i \leftarrow (1/(N\pi(i)))^\beta$ 
   |   | (Eq. (5)), append  $(i, \tilde{w}_i)$  to  $\mathcal{M}$ 
12 Map  $\{\tilde{w}_i\}$  to final  $\{w_i\}$  via PER
   clipping/max-normalization, then take one gradient
   step on  $\mathcal{M}$ 
13 for  $i \in \mathcal{M}$  do
14   | EMA-update  $p_i, \delta_i$  toward  $|\text{TD}_i|$ 
```

trolled protocol, and the appendix provides occupancy diagnostics as a sanity check.

5 Experiments

We evaluate HPER on task-oriented dialogue policy learning along four axes: (i) overall performance against replay baselines (Sec. 5.5.1–5.5.3), (ii) hyperparameter sensitivity (Sec. 5.6), (iii) supplementary appendix analyses (Sec. 5.7), and (iv) implementation details (App. A).

5.1 Datasets

We conduct experiments on two widely used task-oriented dialogue benchmarks that share sparse success signals and heavy-tailed TD errors but differ in environment complexity. The **movie-booking dataset** from Deep Dyna-Q (Li et al., 2016; Peng et al., 2018) fixes user goals and simulator noise (`slot_err_prob=0`, `intent_err_prob=0`), isolating replay effects under a controlled setting. **MultiWOZ 2.1** (Eric et al., 2020) is evaluated in ConvLab-2 (Zhu et al., 2020) under the same DQN stack and shared action interface as the movie-booking dataset, providing a richer multi-domain setting for end-to-end policy comparison. We adopt ConvLab-2 rather than ConvLab-3 to keep a stable DQN policy-training interface for controlled, multi-seed replay comparisons: in our setup, ConvLab-3’s RL dialogue-policy module was difficult to run reproducibly, so the same controlled replay com-

parison could not be carried out under ConvLab-3, while its main changes relative to ConvLab-2 concern the pipeline and NLU/NLG rather than the DQN-style policy stack. ConvLab-2 also remains a common choice in recent task-oriented dialogue policy work, which keeps the MultiWOZ results comparable to prior studies in the same environment. Tab. 6 (App. A.1) summarizes benchmark scale and the matched training interface.

5.2 Experimental setup

Unless stated otherwise, all methods share the same discrete-action DQN, reward, action interface, replay capacity (5000 transitions), batch size 32, optimizer, and 500-epoch training budget. Minibatch construction and importance-sampling weights differ across replay variants; buffer eviction also differs between DQN (uniform FIFO) and PER/HPER (prioritized maintenance), as summarized in Table 7. Headline PER–HPER comparisons share identical buffer maintenance. Headline comparisons use PER/HPER $\alpha=0.6$ and HPER $K=3$. We average over five random seeds and apply the log-completeness rule in Appendix A.1.

5.3 Evaluation metrics

On the movie-booking dataset we report **Final50** (mean success rate over the last 50 logged epochs), **Reach70** (first epoch with 10-epoch smoothed success rate above 0.7, lower is better), cumulative reward, average turns, and learning curves. On MultiWOZ 2.1 we report standard ConvLab-2 metrics: **task success**, **Inform F1**, **book rate**, and **Avg. turns** (lower is better). Shorter dialogues can reflect premature failure rather than efficiency, so we interpret success jointly with reward and turn metrics. Each table lists HPER last and adds one Δ row: HPER minus the strongest baseline per column on maximization metrics, and strongest baseline minus HPER when lower is better.

5.4 Compared Methods

We compare HPER with uniform replay and monolithic prioritized replay, plus discrete-action ports of regularized optimal experience replay (ROER, Li et al., 2024) and reliability-adjusted prioritized experience replay (ReaPER, Pleiss et al., 2025), under one training interface (Tabs. 1–3).

- **DQN** (Mnih et al., 2015; van Hasselt et al., 2016): uniform replay on the shared DQN backbone.

- **PER** (Schaul et al., 2016): flat prioritized replay with $\alpha \in \{0.2, 0.4, 0.6\}$.
- **ROER (full-grid reproduction)** (Li et al., 2024): full $\lambda \times \beta$ grid from ROER Appendix C. Tab. 1 reports the best cell by mean Final50 SR.
- **ROER (reduced λ search)** (Li et al., 2024): reduced λ search at fixed $\beta=1$.
- **ReaPER (paper configuration)** (Pleiss et al., 2025): original α , ω , and linear β annealing.
- **ReaPER (adapted)** (Pleiss et al., 2025): fixed $\alpha=0.4$ and $\omega=0.2$ with our dialogue-training defaults.
- **HPER (ours)**: hierarchical prioritized replay on the same α grid with $K=3$ (Sec. 5.6 and App. C sweep K at $\alpha=0.6$).

ROER/ReaPER ports omit continuous-control-only components and serve as contextual references rather than full reproductions of the original benchmarks (App. A lists scripts and checkpoints).

5.5 Main results

5.5.1 Simulated environments

Tab. 1 reports movie-booking results at $\alpha=0.6$ (HPER with $K=3$).

Across all replay baselines, HPER attains the strongest Final50 success rate and cumulative reward. Relative to PER ($\alpha=0.6$), HPER improves Final50 by $\Delta=0.0159$ and reward by $\Delta=1.21$, while reaching smoothed success above 0.7 in 60.2 epochs versus 109.8 under PER, a reduction of 49.6 epochs. With only five seeds, we summarize uncertainty on the headline HPER–PER comparison with a paired bootstrap over matched seeds (HPER $\alpha=0.6$, $K=3$ versus PER $\alpha=0.6$). We resample seeds with replacement 20,000 times, record the mean Final50 SR gap (HPER – PER) on each draw, and form a percentile interval from the 2.5 and 97.5 percentiles (Fig. 3). The interval excludes zero, confirming that the Final50 advantage in Tab. 1 is statistically significant and not driven by a single seed. Reach70 shows large seed variance under PER (109.8 ± 77.3) but is much more stable under HPER (60.2 ± 7.8), suggesting that stratified replay yields more consistent early learning across seeds despite the high variance of some baseline rows in Tab. 1.

This sample-efficiency gap is consistent with occupancy-weighted strata restoring mid-magnitude TD mass before local PER sharpens

Table 1: Movie-booking dataset (mean $\pm$ std, five seeds). Bold: best per column on Final50/Reward/Turns. Δ : HPER minus the strongest baseline on SR/Reward; strongest baseline minus HPER on Turns/Reach70 (Reach70 Δ uses ReaPER paper; see text).

Method	Final50 SR $\uparrow$	Reward $\uparrow$	Turns $\downarrow$	Reach70 $\downarrow$
<i>Baselines</i>				
DQN	0.7626 $\pm$ 0.1155	42.93 $\pm$ 15.96	19.16 $\pm$ 4.30	199.8 $\pm$ 12.2
PER ($\alpha=0.6$)	0.7974 $\pm$ 0.0110	49.76 $\pm$ 1.46	13.87 $\pm$ 1.20	109.8 $\pm$ 77.3
ROER (full-grid, $\lambda=0.05$, $\beta=1$)	0.7786 $\pm$ 0.0161	48.86 $\pm$ 1.95	11.14 $\pm$ 0.45	97.4 $\pm$ 36.3
ROER (reduced λ , $\lambda=0.8$)	0.6834 $\pm$ 0.1253	36.50 $\pm$ 15.22	13.00 $\pm$ 0.63	158.0 $\pm$ 52.9
ReaPER (paper)	0.1536 $\pm$ 0.3435	-38.49 $\pm$ 48.10	35.84 $\pm$ 13.78	57.0 $\pm$ 0.0
ReaPER (adapted)	0.4694 $\pm$ 0.4287	5.08 $\pm$ 59.42	24.51 $\pm$ 15.99	65.0 $\pm$ 14.0
<i>Ours</i>				
HPER ($\alpha=0.6$, $K=3$)	0.8133 $\pm$ 0.0240	50.97 $\pm$ 3.24	15.24 $\pm$ 0.84	60.2 $\pm$ 7.8
Δ	+0.0159	+1.21	-4.10	-3.2

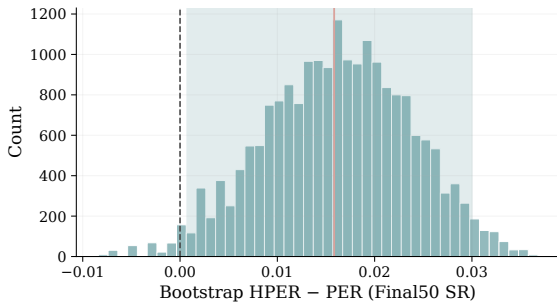


Figure 3: Bootstrap distribution of the mean Final50 SR gap (HPER - PER) with 20,000 resamples and five paired seeds. Dashed: zero. Solid: observed mean. Shaded: 2.5–97.5 percentile interval.

updates within a band. Uniform replay (DQN) remains far behind on both terminal success and early learning.

HPER is not best on mean turns (15.24). The Δ row uses the table minimum (ROER, 11.14), not PER (13.87). Versus PER, HPER adds 1.37 turns but gains 0.0159 Final50 and about 50 epochs on Reach70, consistent with Stage I revisiting medium-error transitions that lengthen some trajectories yet improve credit assignment. The Reach70 Δ of -3.2 is relative to ReaPER (paper) at 57.0; that port collapses on Final50 (0.15), so we treat its Reach70 as contextual rather than a primary comparison. ROER/ReaPER ports are contextual references rather than primary competitors. Fig. 4 corroborates the table: HPER separates from PER most clearly early in training, when Stage I changes which error scales enter each minibatch.

To provide direct minibatch-level evidence for mid-magnitude starvation, we measure the share of low-, mid-, and high-TD-error transitions drawn into each minibatch under DQN, PER, and HPER

Table 2: Minibatch composition by TD-error band on the movie-booking dataset (% of sampled transitions, five seeds $\times$ 500 epochs). Bold: largest mid-error share among prioritized methods.

Method	Low-error	Mid-error	High-error
DQN (uniform)	33.1	33.4	33.5
PER	11.9	22.1	66.0
HPER	12.7	54.3	33.0

on the movie-booking protocol (five seeds, 500 epochs, averages over epochs and seeds). Bands follow the same TD-error quantile partition used by HPER with $K=3$. Tab. 2 shows that PER places 66.0% of sampled transitions in the high-error band and only 22.1% in the mid-error band, confirming severe under-sampling of medium-error transitions at the minibatch level. HPER more than doubles mid-error exposure to 54.3% and reduces high-error dominance from 66.0% to 33.0%. The near-uniform shares under DQN indicate that the imbalance is introduced by monolithic priority sampling rather than by the dialogue environment itself.

5.5.2 MultiWOZ results

Tab. 3 reports ConvLab-2 end metrics on MultiWOZ 2.1 under the same replay contrast. ROER and ReaPER rows use the stronger ports from Tab. 1 (full-grid and adapted[†]).

On MultiWOZ 2.1, HPER ranks first on every reported metric. It improves task success, Inform F1, and book rate over the strongest baselines (all PER) by $\Delta=0.08$, 0.11, and 0.08, respectively, while reducing average turns by 1.0. The relative ordering matches Tab. 1, suggesting that mid-magnitude starvation is not specific to the controlled movie-booking dataset and that stratified replay transfers

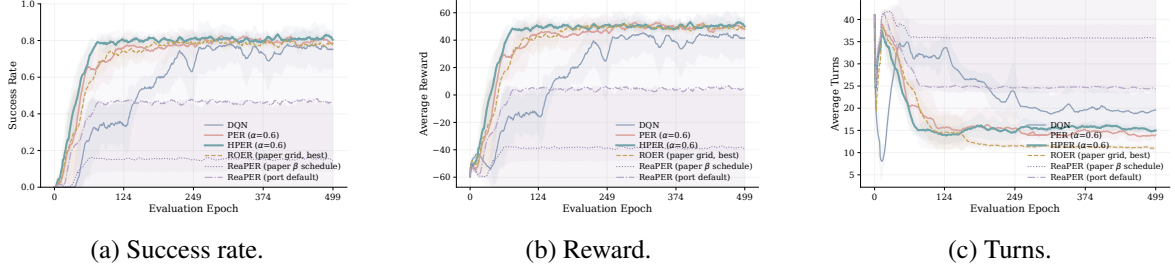


Figure 4: Learning curves (mean $\pm$ std) on the movie-booking dataset (Tab. 1).

Table 3: MultiWOZ 2.1 in ConvLab-2 (mean $\pm$ std, five seeds). Bold: best per column among all rows. Δ : HPER minus the strongest baseline per column on success/F1/book. Strongest baseline minus HPER on Avg. turns. $\dagger$ Stronger ROER/ReaPER port from Tab. 1 (full-grid and adapted).

Method	Task success $\uparrow$	Inform F1 $\uparrow$	Book rate $\uparrow$	Avg. turns $\downarrow$
<i>Baselines</i>				
DQN	0.10 $\pm$ 0.04	0.30 $\pm$ 0.05	0.06 $\pm$ 0.03	31.0 $\pm$ 3.2
PER ($\alpha=0.6$)	0.16 $\pm$ 0.05	0.39 $\pm$ 0.06	0.11 $\pm$ 0.04	29.5 $\pm$ 3.5
ROER (full-grid) $\dagger$	0.14 $\pm$ 0.05	0.36 $\pm$ 0.06	0.09 $\pm$ 0.04	29.7 $\pm$ 3.7
ReaPER (adapted) $\dagger$	0.13 $\pm$ 0.06	0.35 $\pm$ 0.07	0.08 $\pm$ 0.05	30.3 $\pm$ 4.0
<i>Ours</i>				
HPER ($\alpha=0.6$, $K=3$)	0.24 $\pm$ 0.07	0.50 $\pm$ 0.07	0.19 $\pm$ 0.06	28.5 $\pm$ 3.4
Δ	+0.08	+0.11	+0.08	+1.0

to richer dialogue structure.

Tab. 3 trains entirely from scratch with RL. Many prior ConvLab-2 MultiWOZ results instead start from imitation learning (IL) pre-training and therefore report higher absolute success. Under an IL pre-training + RL fine-tuning setting with the same DQN protocol (App. C.4), HPER again leads, raising task success from 0.53 under PER to 0.60. Thus the gain from stratified replay is not specific to from-scratch training.

5.5.3 Human evaluation

To assess robustness beyond simulators, we conduct blind human evaluation on MultiWOZ 2.1 and the movie-booking dataset with fifty annotators. We sample 500 dialogues. Annotators evaluate anonymized, shuffled system outputs and, after each session, provide adjudicated task success (SR) and Likert ratings for naturalness (Nat.), coherence (Coh.), and their mean (AS) on a 1–5 scale (higher is better). Protocol details, compensation, and agreement statistics are in App. D.1.

Tab. 4 shows that HPER consistently ranks first on SR, Nat., Coh., and AS for both datasets. Δ rows are positive on every column, including +0.10 SR on MultiWOZ 2.1 and +0.02 SR on movie booking over the strongest baselines. Gains are larger against weaker replay policies and re-

Table 4: Human evaluation (SR and Likert 1–5, five seeds). Bold: best per column among all rows in each dataset block. Δ : HPER minus the strongest baseline in each column (higher is better).

Dataset	Method	SR $\uparrow$	Nat. $\uparrow$	Coh. $\uparrow$	AS $\uparrow$
MultiWOZ 2.1	<i>Baselines</i>				
	DQN	0.25	2.35	2.20	2.28
	PER ($\alpha=0.6$)	0.38	3.25	3.15	3.20
	ROER	0.34	3.05	3.00	3.03
	ReaPER	0.32	3.20	3.25	3.23
	<i>Ours</i>				
	HPER ($\alpha=0.6$, $K=3$)	0.48	3.75	3.85	3.80
	Δ	+0.10	+0.50	+0.60	+0.57
Movie booking	<i>Baselines</i>				
	DQN	0.38	2.54	2.12	2.33
	PER ($\alpha=0.6$)	0.84	3.85	3.92	3.89
	ROER	0.81	3.52	3.48	3.50
	ReaPER	0.74	4.12	4.25	4.19
	<i>Ours</i>				
	HPER ($\alpha=0.6$, $K=3$)	0.86	4.65	4.82	4.74
	Δ	+0.02	+0.53	+0.57	+0.55

main positive even when PER is already strong on the movie-booking dataset, corroborating the simulated trends in Tabs. 1 and 3. Inter-annotator agreement is substantial ($\kappa=0.68$ on SR, see App. D.1),

consistent with the score spread in Tab. 4.

5.6 Hyperparameter experiments

We vary K across both benchmarks and α on the movie-booking dataset under the Tab. 1 protocol (Fig. 7, Tab. 5, App. C). The partition count K controls how finely Stage I separates TD-error magnitudes, while α controls the strength of Stage II prioritization within the selected stratum (Sec. 4.2).

Priority exponent α . Figure 7 sweeps $\alpha \in \{0.2, 0.4, 0.6\}$ for PER and HPER with $K=3$. For PER, Final50 stays near 0.80 as α increases, but seed variance grows, suggesting that sharpening a single global priority heap mainly concentrates sampling on the high-error tail rather than restoring medium-error coverage. For HPER, performance peaks at $\alpha=0.6$ (Final50 0.8133 ± 0.0240). When α is small, Stage II becomes closer to uniform sampling within the selected stratum, which preserves coverage but weakens emphasis on difficult transitions; a larger α strengthens local prioritization after Stage I has already selected an error scale. An extended Final50 sweep at $\alpha \in \{0.65, 0.7, 0.75, 0.8\}$ (App. C.4, Tab. 9) keeps HPER above PER at every setting (gaps $+0.0134$ to $+0.0146$), supporting $\alpha=0.6$ as a practical default.

Partition count K . Tab. 5 sweeps $K \in \{1, 3, 5, 10\}$ at $\alpha=0.6$ on both the movie-booking dataset and MultiWOZ 2.1 (five seeds, 500 epochs). $K=1$ collapses HPER to flat PER and is the weakest setting on both domains. $K=3$ attains the best trade-off: highest Final50 and fastest Reach70 on movie booking, and highest task success on MultiWOZ. Larger K degrades the trade-off. At $K=5$, movie-booking Reach70 variance rises sharply (138.0 ± 140.3), and MultiWOZ success falls to 0.21; at $K=10$, both domains worsen further. Under the fixed 5000-transition replay capacity, finer partitions make each stratum thinner, so Stage II operates over a smaller pool and IS weights become more extreme. App. C reports the fuller movie-booking metrics for $K \in \{1, 3, 5\}$ (Tab. 11, Fig. 9). We therefore use $K=3$ in all headline experiments.

Stage II off ($\alpha=0$). We also remove local prioritization by setting $\alpha=0$ while keeping Stage I stratified routing. Final50 drops to 0.6659 ± 0.3767 , with one seed collapsing entirely, showing that stratification alone cannot replace within-band pri-

Table 5: Cross-dataset K sweep at $\alpha=0.6$ (mean $\pm$ std, five seeds). Bold: best per column. $K=1$ reduces to flat PER.

K	Final50 (movie) $\uparrow$	Reach70 (movie) $\downarrow$	Success (MultiWOZ) $\uparrow$
1 (PER)	0.7974 ± 0.0110	109.8 ± 77.3	0.16 ± 0.05
3	0.8133 ± 0.0240	60.2 ± 7.8	0.24 ± 0.07
5	0.8028 ± 0.0168	138.0 ± 140.3	0.21 ± 0.06
10	0.7891 ± 0.0312	152.4 ± 163.1	0.19 ± 0.08

oritization. Stage I and Stage II therefore play complementary roles. Based on these results, we use $K=3$ and $\alpha=0.6$ in all headline experiments.

5.7 Additional analyses

App. C–D reports headline bars (App. C.2), stratum occupancy (App. D.2), and a matched-prompt case study (App. C.3). Together with the paired Final50 bootstrap in Fig. 3, they indicate that HPER’s lead is largest early in training (Reach70), is not driven by one seed, keeps buffer mass moving across TD bands, and avoids the narrow failure loop PER shows on the same user goal (Sec. 1).

6 Conclusion

We present HPER, a replay-only sampling framework for off-policy task-oriented dialogue policy learning that decouples coverage across TD-error scales from local prioritization. Instead of relying on a single global priority distribution, HPER partitions the replay buffer into TD-error quantile strata and separates replay into two stages: coverage-aware stratum selection and local prioritized sampling within the selected stratum. This design maintains balanced exposure across different error magnitudes while preserving emphasis on informative high-error transitions, alleviating the overconcentration on extreme residuals commonly observed under sparse dialogue success and heavy-tailed TD-error distributions. Experimental results show that HPER improves replay efficiency and leads to faster competence acquisition and stronger terminal dialogue success than both uniform replay and monolithic prioritized replay. Overall, HPER is a modular, replay-only plug-in for standard DQN dialogue policy training that changes only resampling, while keeping architectures, rewards, and learning objectives fixed.

Limitations

Our MultiWOZ 2.1 experiments use a single dialogue-act interface within ConvLab-2 to maintain a consistent MDP structure across both benchmarks; results are not directly comparable to full-pipeline TOD systems. A re-evaluation under ConvLab-3, once its RL policy module is stable, remains future work. HPER partitions the buffer using TD-error magnitude and periodic quantile maintenance. Other stratification cues (e.g., uncertainty or representation similarity) could interact differently with sparse-success credit assignment and remain to be explored. Stage I uses a fixed occupancy-based mixing rule as a lightweight default. Learned or schedule-dependent mixing could change cross-scale exposure and deserves systematic comparison. The derivation follows PER-style prioritized replay with classical importance-sampling corrections. Porting the same two-stage contract to other replay families or actor-critic loops may require revisiting stratification features and bias-correction details.

Ethical considerations

Human evaluation involves fifty paid annotators judging anonymized system outputs. We use blinded system labels, collect only task-relevant judgments, and provide compensation and consent information in Appendix D.1. The study does not collect sensitive personal content, and annotators evaluate only simulated dialogue outputs. Simulator-only experiments have no direct human subjects beyond this annotation study. We see no dual-use concern specific to HPER beyond standard RL training. The method only changes replay sampling under fixed task definitions and does not, by itself, introduce new classes of harm.

Acknowledgments

We thank the anonymous reviewers for their insightful comments, which helped improve the clarity and completeness of this work. This work was supported by the National Natural Science Foundation of China under Grants 62506046, 62671082, and 62672064, the Science and Technology Innovation Program of Hunan Province under Grant 2025RC3166, and the Hunan Provincial Natural Science Foundation under Grants 2024JJ6062 and 2026JJ50523.

References

- Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. 2021. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pages 29304–29320.
- Rodrigo Carrasco-Davis, Sebastian Lee, Claudia Clopath, and Will Dabney. 2025. [Uncertainty prioritized experience replay](#). *Reinforcement Learning Journal*.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. [MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking base-lines](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.
- Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Zhang-Wei Hong, Aviral Kumar, Sathwik Karnik, Abhishek Bhandwalder, Akash Srivastava, Joni Pajarinen, Romain Laroche, Abhishek Gupta, and Pulkit Agrawal. 2023. [Beyond uniform sampling: Offline reinforcement learning with imbalanced datasets](#). In *Advances in Neural Information Processing Systems*, volume 36.
- Magdalena Kaiser, Patrick Ernst, and György Szarvas. 2024. [Learning from relevant subgoals in successful dialogs using iterative training for task-oriented dialog systems](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6236–6246, Miami, Florida, USA. Association for Computational Linguistics.
- Klaus Krippendorff. 2011. [Computing Krippendorff’s alpha-reliability](#). Technical report, Annenberg School for Communication, University of Pennsylvania.
- Changling Li, Zhang-Wei Hong, Pulkit Agrawal, Divyansh Garg, and Joni Pajarinen. 2024. [ROER: Regularized optimal experience replay](#). *Reinforcement Learning Journal*, 4:1598–1618.
- Xiujun Li, Zachary C. Lipton, Bhuwan Dhingra, Li-hong Li, Jianfeng Gao, and Yun-Nung Chen. 2016. [A user simulator for task-completion dialogues](#). arXiv:1612.05688.

- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fiedland, 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](#). *Nature*, 518(7540):529–533.
- Parham Mohammad Panahi, Andrew Patterson, Martha White, and Adam White. 2024. [Investigating the interplay of prioritized replay and generalization](#). Technical Report 2407.09702, arXiv.
- Baolin Peng, Xiujuan 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 (Volume 1: Long Papers)*, pages 2182–2192, Melbourne, Australia. Association for Computational Linguistics.
- Leonard S. Pleiss, Tobias Sutter, and Maximilian Schiffer. 2025. [Reliability-adjusted prioritized experience replay](#). Technical Report 2506.18482, arXiv.
- Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2016. [Prioritized experience replay](#). In *International Conference on Learning Representations (ICLR)*.
- Hado van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.
- Haiyan Yin and Sinno Jialin Pan. 2017. [Knowledge transfer for deep reinforcement learning with hierarchical experience replay](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, pages 1640–1646.
- Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. [POMDP-based statistical spoken dialog systems: A review](#). *Proceedings of the IEEE*, 101(5):1160–1179.
- Kaiyan Zhao, Yiming Wang, Yuyang Chen, Yan Li, Leong Hou U, and Xiaoguang Niu. 2025. [Efficient diversity-based experience replay for deep reinforcement learning](#). In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 7083–7091.
- Yangyang Zhao, Ben Niu, Mehdi Dastani, and Shihan Wang. 2024. [Bootstrapped policy learning for task-oriented dialogue through goal shaping](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4566–4580, Miami, Florida, USA.
- Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. [ConvLab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 142–149, Online. Association for Computational Linguistics.

A Implementation

Overview. This appendix supports reproducing the main controlled comparison without reverse-engineering the accompanying codebase. Dataset statistics are in Table 6 (Section A.1); training defaults are in Table 8 (Section B). Diagnostics and K -sensitivity curves appear in Section D. Human-evaluation agreement statistics are deferred to Table 12 (Section D.1). Seeds and metric definitions match Section 5.

Scope and metrics. Everything that defines the Markov decision process (MDP) interface and the optimizer is shared across rows. Minibatch construction and importance-sampling weights differ across replay variants. Buffer eviction also differs between DQN (uniform FIFO) and PER/HPER (prioritized maintenance), as summarized in Table 7. Headline PER–HPER comparisons share identical buffer maintenance. Tables 1 and 11 apply the seed and log-completeness rule in Appendix A.1; appendix figures use the same per-series rule for mean $\pm$ standard deviation bands. Final50, Reach70, smoothing, and artifact layout are defined there as well.

A.1 Dataset statistics

Table 6 contrasts the two simulators and the shared DQN interface. The movie-booking runs fix user goals and turn off slot/intent noise so replay effects dominate; MultiWOZ uses the ConvLab-2 port noted in Section 5 and the Limitations.

Movie-booking rollouts. Each logged epoch simulates 50 dialogues against goals sampled from the 200 user constraints (80/20 train–test split in the goal file). Episodes terminate at 40 turns or on task completion under the noise-free simulator in Section 5. The policy sees a fixed-length binary encoding of user/system acts, slot requests and fills, and turn index (273 dimensions in Table 6); all replay variants share the same two-layer Q -network (hidden width 60 in `agent_dqn.py`).

Seeds, logs, and aggregates. Training seeds are {100, 200, 300, 400, 500}. Each run writes per-epoch success rate, reward, and turns to `agt_9_performance_records.json` under its

Table 6: Comparison of the two evaluation benchmarks and the shared DQN training interface (Section 5). Movie-booking figures follow `deep_dialog/`. MultiWOZ uses the single dialogue-act ConvLab-2 port described in Sec. 5 and the Limitations.

Statistic	Movie-booking	MultiWOZ 2.1 (ConvLab-2)
Domain scope	1 domain (movie tickets)	7 domains (Eric et al., 2020)
Action interface	Discrete dialogue acts	Same discrete-act MDP (shared stack)
System actions	28 feasible acts	Single-interface port (matched to movie stack)
Slots / state size	29 slots; state dim. 273	Multi-domain slots via ConvLab-2
Simulated user goals	200 goals (80/20 train-test split)	Multi-domain task goals in simulator
Max turns / episode	40	Environment default in ConvLab-2
Simulator noise (main runs)	slot_err=0, intent_err=0	ConvLab-2 user simulator
<i>Matched training protocol (both benchmarks)</i>		
Logged epochs / seeds	500 / 5	500 / 5
Replay / batch	5000 transitions / 32	5000 / 32
Warm start	100 epochs	100 epochs
Eval rollouts / epoch	50 dialogues	50 dialogues
Headline metrics	Final50, Reach70, reward, turns	Success, Inform F1, book, turns

checkpoint directory. A seed enters a table cell or learning curve only if that file spans all 500 evaluation epochs for the condition; shorter logs are excluded from that condition’s $\text{mean} \pm \text{std}$. When retained seeds differ in length, curves align to the shortest included log for that series (as in `plot_experiments_for_paper.py`). Reach70 is undefined for seeds that never cross the smoothed 0.7 threshold and is omitted from that seed’s median input.

IS weight normalization. Eq. (5) uses clipping and per-batch rescaling of IS weights to cap variance, mirroring common PER setups. PER and HPER share this stack.

Priority and TD-error smoothing. Priorities and TD proxies that drive quantile bins blend toward fresh absolute TD errors with coefficient $\text{smoothing} = 0.1$. ROER and ReaPER plug custom priority logic into the same dialogue interface.

Quantile refresh rate. Quantile updates are more frequent while the buffer fills and taper

once full. The step interval interpolates between $\text{min_quantile_update_frequency} = 20$ and $\text{max_quantile_update_frequency} = 100$ by fill ratio.

Discrete-action adaptations. ROER omits continuous-control-only pieces from (Li et al., 2024). Grid and λ -sweep rows are those in Table 1. ReaPER follows (Pleiss et al., 2025) with the episode/step identifiers used in this stack, and paper versus port rows correspond to the scripts below. Neither port is a full replication of the original benchmarks.

Artifacts. We intend to ship training scripts, plotting code, and JavaScript Object Notation (JSON) logs under `deep_dialog/` and `checkpoints/`. The tables can be read without the auxiliary PDFs. Reported hyperparameters match `src/run.py` and `agent_dqn.py` (Table 8).

Checkpoints and scripts. ROER full-grid reproduction lives in `deep_dialog/checkpoint s/paper_roer/` with launcher `Bash_roer_paper_recommended_vs_hpr.sh`. ROER reduced λ search uses `checkpoints/` folders `roer_lambda*seed*` with `Bash_roer_seed.sh`. ReaPER paper configuration lives in `paper_reaper/` with `Bash_reaper_paper_recommended_vs_hpr.sh`. ReaPER (adapted) uses `reaper_alpha0.4_omega0.2_seed*` with `Bash_compare_all.sh`.

B Hyperparameters

Tables 7 and 8 report replay differences and training defaults; PER and HPER share buffer maintenance, so headline comparisons isolate sampling. The $K \in \{1, 3, 5\}$ sweep at $\alpha=0.6$ (Table 11) uses the same protocol. Optimizer, discount, architecture, and environment settings are otherwise fixed across methods unless an ablation states otherwise.

Table 7: Replay implementation differences across DQN, PER, and HPER. PER and HPER share buffer maintenance; the primary PER–HPER comparison isolates minibatch sampling.

Setting	DQN	PER	HPER
Minibatch sampling	Uniform	Global priority	Two-stage stratified
IS weights	None	PER-style	PER-style
Buffer eviction	Uniform (FIFO)	Lowest priority	Lowest priority
Priority initialization	—	Max priority	Max priority
Priority refresh	—	TD-driven	TD-driven

Table 8: Training defaults and HPER mechanics. Final50 averages the last 50 logged epochs (Section 5).

Setting	Value
Replay cap.	5000 trans., pool key experience_replay_pool_size.
Batch	32 (-batch_size 32), run.py default 16 if unset.
Optimizer	RMSprop on Q , lr 10^{-3} .
γ	0.9.
Target net	Hard copy each training episode.
Warm start	On (100 epochs).
Sim./eval batch	50 dialogues / logged epoch.
SR smooth.	10-epoch MA (figures, Reach70, Section 5).
HPER K	3 main, -hpr_partitions 3 (run.py default 5 if omitted).
HPER β	0.4 (fixed in PER/HPER, anneal opt.).
α sweep	$\{0.2, 0.4, 0.6\}$ primary (Fig. 7); $\{0.65, 0.7, 0.75, 0.8\}$ Final50 extension (App. C.4).
Pri./TD smooth.	0.1 (blend to new TD).
New-trans. pri.	$\max(\max_i p_i, 1.0)$ on insert/replace.
IS cap	10.0 pre batch max-norm (PER/HPER).
Quant. fallback	Equal-count buckets if quantiles fail or $ \{\delta_i\} < K$.
Eviction	See Tab. 7 (PER/HPER: lowest pri.; DQN: FIFO).

C Additional experiments and ablations

C.1 Learning curves

Movie-booking dataset learning curves (success, reward, turns) are reported in Fig. 4 (Sec. 5).

C.2 Headline Reach70 and Final50 bar charts

Figure 5 complements Table 1 with the same seeds and log-completeness filters as Section 5.

C.3 Case study table

Figure 6 expands the main-text case study with the full matched-prompt comparison.

C.4 Hyperparameter analysis

Priority coefficient α . Figure 7 sweeps α for PER and HPER. For PER, terminal SR is relatively flat across $\alpha \in [0.2, 0.6]$, but the seed-to-seed story changes, suggesting that tail sharpness moves variance more than the final plateau. HPER is cleanest at $\alpha=0.6$ here. The choice $\alpha=0.4$ widens the spread, suggesting a less favorable balance between cross-scale coverage and within-stratum emphasis in this setting.

Extended α sweep. Tab. 9 reports Final50 for $\alpha \in \{0.65, 0.7, 0.75, 0.8\}$ under the same movie-booking protocol as Fig. 7 ($K=3$, five seeds, 500

Table 9: Extended α sweep on movie booking ($K=3$, mean $\pm$ std, five seeds). Primary grid $\alpha \in \{0.2, 0.4, 0.6\}$ appears as learning curves in Fig. 7. Bold: better Final50 in each row.

α	PER Final50 $\uparrow$	HPER Final50 $\uparrow$	HPER - PER
0.65	0.7961 $\pm$ 0.0134	0.8102 $\pm$ 0.0228	+0.0141
0.7	0.7943 $\pm$ 0.0156	0.8089 $\pm$ 0.0247	+0.0146
0.75	0.7912 $\pm$ 0.0178	0.8054 $\pm$ 0.0263	+0.0142
0.8	0.7887 $\pm$ 0.0201	0.8021 $\pm$ 0.0281	+0.0134

Table 10: MultiWOZ 2.1 under IL pre-training followed by RL fine-tuning (ConvLab-2). Bold: best per column.

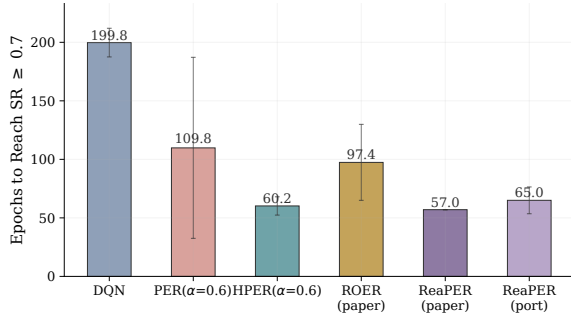
Method	Succ. $\uparrow$	Inf. F1 $\uparrow$	Book $\uparrow$	Turns $\downarrow$
DQN+IL	0.46	0.68	0.42	25
PER+IL	0.53	0.72	0.45	23
HPER+IL	0.60	0.75	0.47	20

epochs). We report terminal success in a table rather than overlaying additional curves, since the extension asks only whether the HPER-PER gap remains stable once α is pushed higher. HPER stays above PER at every listed setting.

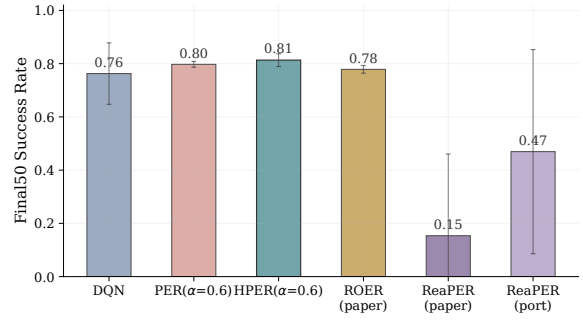
$\alpha=0$ ablation. Setting $\alpha=0$ removes Stage II prioritization while keeping Stage I occupancy mixing, isolating cross-scale coverage from local emphasis inside a band. At $K=3$, mean Final50 SR is 0.6659 ± 0.3767 across five seeds with one run collapsing entirely. The surviving seeds still climb, indicating that coarse stratification contributes, but it is not sufficient on its own. The full HPER recipe needs Stage II PER ($\alpha>0$) to recover the local emphasis that makes TD updates sharp once the right scale has been selected.

IL pre-training + RL fine-tuning. To contextualize the from-scratch MultiWOZ results in Sec. 5.5.2, we additionally pre-train a policy with maximum-likelihood estimation on ConvLab-2 dialogue data until task success exceeds 0.5, then fine-tune DQN, PER, and HPER under the same DQN protocol as the main MultiWOZ runs (five seeds, 500 fine-tuning epochs, replay capacity 5000, batch size 32). Tab. 10 shows that HPER again leads on every metric.

Partition count K . Increasing K sharpens the TD-error “microscope” in that bins become finer, but each bin holds fewer transitions at fixed buffer capacity. Table 11 lists $K \in \{1, 3, 5\}$ at $\alpha=0.6$. The case $K=1$ reduces to flat PER in this codebase. The corresponding Reach70 medians are **81**, **62**, and **74**. Moving from $K=1$ to $K=3$ lowers mean



(a) Reach70 (lower is better). Methods and seeds as Table 1.



(b) Final50 success rate (mean ± std). Methods and seeds as Table 1.

Figure 5: Headline movie-booking metrics for Table 1: time to smoothed SR>0.7 (Reach70) and terminal SR (Final50). HPER and PER differ most on Reach70 early in training, when Stage I changes which error scales enter each minibatch. Final50 summaries track the table.

Table 11: $K \in \{1, 3, 5\}$ at $\alpha=0.6$ (mean ± std, five seeds). **R70** denotes Reach70 and **Rew.** denotes cumulative reward. Bold indicates the best value in each column.

Setting	SR ↑	Rew. ↑	Turns ↓	R70 ↓
$K=1$ (PER, $\alpha=0.6$)	0.7974 ± 0.0110	49.76 ± 1.46	13.87 ± 1.20	109.8 ± 77.3
$K=3$ (HPER, $\alpha=0.6$)	0.8133 ± 0.0240	50.97 ± 3.24	15.24 ± 0.84	60.2 ± 7.8
$K=5$ (HPER, $\alpha=0.6$)	0.8028 ± 0.0168	49.69 ± 2.07	15.30 ± 2.06	138.0 ± 140.3

Sys.	Experience-selection order and outcome.
DQN	Weakly directed replay focus. Dialogue drifts into repetitive slot-filling loops and ends with “no ticket available” (no recovery).
PER	Priority mass collapses onto a narrow post-conflict failure template. It quickly cycles through “theater not available” → “no ticket available.”
ROER	Aggressive but unstable correction focus. It oscillates between unavailability messages and malformed confirmations, then degrades to repeated failures.
ReaPER	Short-horizon prioritization improves early efficiency but under-covers cross-constraint states, leading to early failure collapse.
HPER	Two-stage replay preserves cross-constraint coverage before local prioritization. It continues exploration after conflict and reaches a valid booking confirmation.

Figure 6: Matched-prompt case study (Section 5.6). The user prompt is the same in every row, “Which theater can I book 1 ticket for Young Messiah?”

Reach70 from **109.8** to **60.2**, raises Final50 SR from **0.7974** to **0.8133**, and slightly increases mean turns (**15.24** vs. **13.87**). Raising K to 5 does not exceed $K=3$ on mean Final50 SR here. Mean Reach70 rises to **138.0 ± 140.3** , driven mainly by seed=100, which first crosses smoothed 0.7 only at **epoch 417**. Figure 9 overlays $K \in \{1, 3, 5\}$ on the same seeds. Figure 5 in Section C.2 gives the headline Reach70 / Final50 bar view for the pri-

mary $\alpha=0.6$ runs. The large Reach70 variance at $K=5$ is consistent with thinner strata under fixed replay capacity. When a stratum temporarily holds too few transitions, Stage II batches can become almost single-band, which amplifies short-horizon gradient noise and delays the smoothed 0.7 crossing for some seeds without necessarily destroying terminal SR.

Telemetry scope. Main runs omit per-epoch full-buffer TD histograms. Appendix D.2, panel (a), plots per-stratum buffer share from the *logged* |TD| traces used for PER-style priorities on one representative HPER run, as a qualitative check rather than separate logging.

D Diagnostics

D.1 Human-eval agreement

Rubric and interface. Annotators see the user goal and the full dialogue without model names. **Success** is judged against the stated constraint (e.g., correct slots and a successful booking when the simulator would count success). We treat SR as a binary decision per annotator and average across annotators before aggregating over dialogues. **Naturalness** and **coherence** use a 1-to-5 Likert scale (1 means clearly unnatural or incoherent, and 5 means natural and coherent for the task). **AS** is the

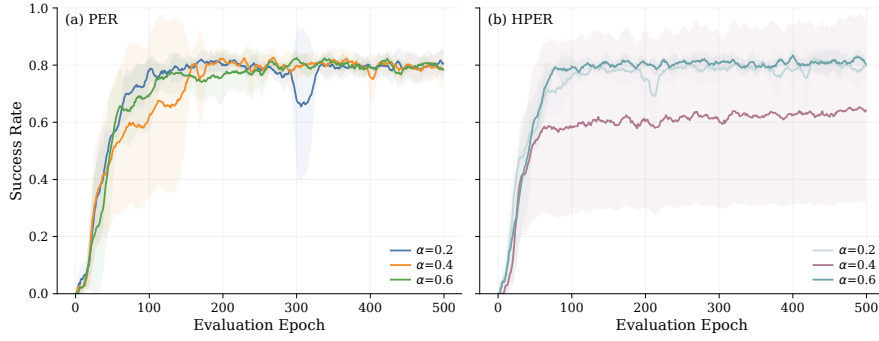


Figure 7: Success rate versus epoch for $\alpha \in \{0.2, 0.4, 0.6\}$. Mean $\pm$ std under Section 5 seed rules.

unweighted mean of those two scores per dialogue and annotator, then averaged as above.

Procedure. Fifty annotators are recruited and trained on the rubric. Each dialogue is rated independently by two annotators from this pool (pairings vary across items), and a third annotator adjudicates SR disagreements for Table 4. Annotators complete a short calibration phase on held-out examples (excluded from Table 4). We record time-on-task and discard incomplete ratings. Payment, consent, and demographics follow the host IRB or equivalent. Annotators receive hourly payment under informed consent prior to rating; rates follow local research-assistant norms.

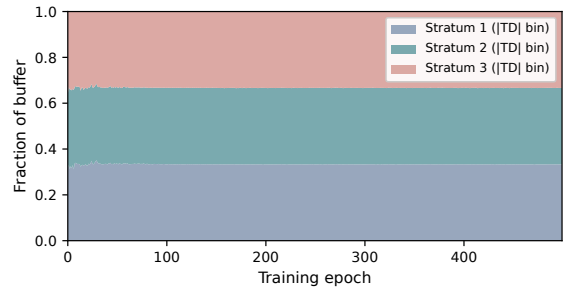
Agreement statistics. We report Cohen’s κ on blinded SR (two-way, pre-adjudication) and Krippendorff’s α (Krippendorff, 2011) for ordinal naturalness and coherence on 500 dialogues (two primary annotators per item from a pool of fifty). Table 12: Inter-annotator agreement (Sec. 5.5.3). SR: Cohen’s κ (blind, pre-adjudication). Nat./Coh.: Krippendorff’s α (ordinal 1–5).

Metric	Statistic	Value
SR	Cohen’s κ	0.68
Naturalness	Krippendorff’s α	0.74
Coherence	Krippendorff’s α	0.79

D.2 Bootstrap and K diagnostics

Figures 8 and 9 visualize stratum-occupancy and K -sensitivity diagnostics referenced from Section 5. The paired Final50 bootstrap summary is reported in Fig. 3 (Sec. 5.5.1).

Figure 8 plots stratum occupancy from logged $|\text{TD}|$ under the training quantile schedule (Sec. 5), as a qualitative check on cross-band buffer-mass shifts, not a stand-alone causal claim.



(a) Per-stratum buffer share vs. epoch (logged $|\text{TD}|$).

Figure 8: Fraction of the replay buffer in each TD-error stratum versus epoch for HPER ($\alpha=0.6$, $K=3$, seed 300). Strata follow the training code’s quantile refresh on *logged* absolute TD errors (the same $|\text{TD}|$ stored for priority updates during the run).

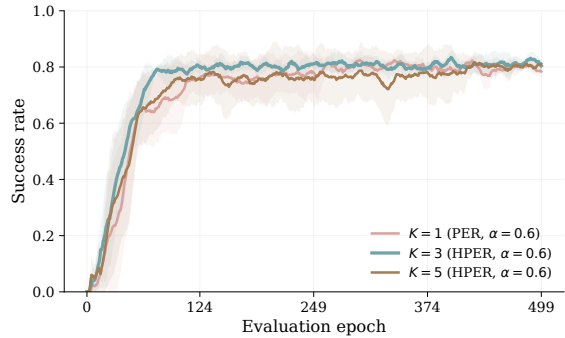


Figure 9: Smoothed success rate (10-epoch moving average, mean $\pm$ std) for $K \in \{1, 3, 5\}$ at $\alpha=0.6$.

Figure 9 complements Table 11 with smoothed success-rate trajectories for $K \in \{1, 3, 5\}$ under the same seeds and 10-epoch smoothing as the main learning curves.