Mitigating Forgetting and Noise: The AMLoss Method for Task-oriented Dialogue Policy Optimization

Anonymous ACL submission

Abstract

Offline Reinforcement Learning (Offline RL) is widely used for optimizing task-oriented dialogue policies by training on pre-collected dialogues, which boosts efficiency, especially when data is limited. However, traditional offline RL methods struggle with accurately measuring experience priority, leading to the loss of valuable data and susceptibility to noisy samples. To this end, this paper proposes the Adjustable Mirror Loss (AMLoss) method, which redefines experience priority by quantifying the real-time incremental contribution of each experience to policy improvement. Specifically, the contribution is computed as the loss difference between the Main and Delayed Q-networks, with a larger difference indicating a more significant learning contribution and, consequently, a higher sampling priority. By emphasizing experiences that offer greater learning gains and deprioritizing those less effective or affected by noise, AMLoss helps retain critical data. Moreover, a Sum Tree structure is introduced for efficient hierarchical storage and weighted sampling of priorities. Experimental results confirm that AMLoss effectively prioritizes important experiences while filtering out noisy ones, leading to optimal performance across various tasks.

1 Introduction

Task-Oriented Dialogue systems (TODs) are designed to accomplish specific tasks, such as booking flights (Algherairy and Ahmed, 2025; Xu et al., 2024; Wang et al., 2023). The core of TODs lies in the design of the Dialogue Policy (DP), which directly influences interaction quality and task completion. Recently, Large Language Models (LLMs) have shown strong performance in open-domain dialogue systems, but they struggle with TODs due to challenges in modeling long-term decision-making (Gao et al., 2024). However, due to data limitations, collecting large amounts of task-oriented dialogue data to fine-tune LLMs is impractical (Chang et al., 2024).

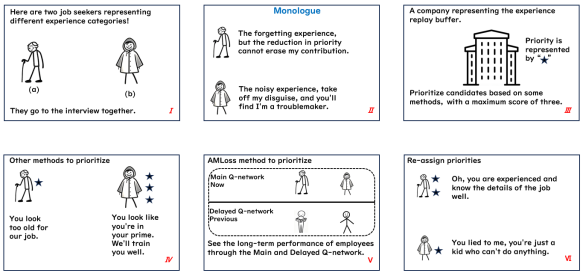


Figure 1: The issue of forgetting and noisy experiences

In contrast, Offline Reinforcement Learning (Offline RL) excels at sequential decision-making tasks, leveraging historical interaction data to reduce reliance on real-time user feedback and improve efficiency, thus becoming a widely used method for DP optimization. (Franceschelli and Musolesi, 2024; Kamuni et al., 2024).

Nevertheless, Offline RL-based DPs face significant challenges of experience priority distortion, resulting in inefficient sampling, particularly in scenarios with insufficient or low-quality samples. This issue primarily arises from two types of experience (Hu et al., 2021; Zhang and Sutton, 2017): **1) Forgetting Experiences:** As shown by example (a) in Figure 1, these experiences appear so outdated that they are often mistakenly deemed unhelpful for DP optimization, resulting in them receiving an extremely low priority (one star). However, such priority allocations overlook the *"experience richness"* inherent in older experiences, which plays a crucial role in the long-term interactions of DP, ultimately leading to suboptimal policy improvement outcomes (Morad et al., 2023; Wang and Ross, 2019). **2) Noisy Experiences:** As shown by example (b) in Figure 1, these experiences are affected by low-quality interactions and errors in upstream modules, which cause them to contain noise. These noises act like a raincoat to obscure the true state of the experience, causing alternative priority assign-

ment methods to mistakenly deem it significant and assign a two-star priority, thereby wasting training resources and compromising the policy updates (Li et al., 2023). Despite extensive efforts to develop mitigation strategies such as experience prioritization weighting (Yu et al., 2024; Vezhnevets et al., 2017; Gu et al., 2017), noise filtering (Yu et al., 2024; Vezhnevets et al., 2017; Gu et al., 2017), and memory enhancement (Lu et al., 2023; Yang et al., 2022; Buzzega et al., 2020), these methods are limited to addressing either the issue of forgetting or noise individually, and none effectively tackle both simultaneously.

To this end, this paper proposes the Adjustable Mirror Loss (AMLoss) method for accurately measuring experience priority to mitigate the impact of noisy and forgetting experiences. Specifically, AMLoss redefines experience priority based on the loss difference between the Main Q-network and the Delayed Q-network. For forgetting experiences, AMloss identifies instances where the Delayed Q-network loss is smaller than the Main Q-network loss. This suggests that their past contributions are greater than their current ones, indicating a richness in experience that significantly benefits DP training. Consequently, AMloss assigns higher priority to these experiences, ensuring they are revisited to restore critical memory and prevent long-term performance degradation. For noisy experiences, AMloss identifies instances where both the Delayed and Main Q-network losses are high, indicating low learning gains over a period. Consequently, AMloss assigns lower priority to these experiences. To enhance the efficiency of weighted sampling based on experience priority, a Sum Tree hierarchical storage structure is introduced. In addition, AMLoss is a model-agnostic method compatible with typical Offline RL approaches that utilize experience replay and target network mechanisms.

Extensive experiments on multiple datasets validate the effectiveness of AMLoss in DP optimization. Furthermore, visualizations of experience priority provide additional insights into how AMLoss manages forgetting and noisy experiences. In summary, our contributions are as follows:

1) We introduce AMLoss to mitigate forgetting and noise in experience replay by dynamically adjusting priorities based on the real-time incremental contribution.

2) We propose a reliable prioritization method that assigns experience priority based on the loss difference between the Main and Delayed Q-networks.

3) We demonstrate the effectiveness and robustness of our approach on four datasets with different noise levels and achieve outstanding performance in human evaluation experiments.

2 Related Work

This paper focuses on enhancing experience replay mechanisms in offline RL to address challenges associated with forgetting and noisy experiences. The foundational method in experience replay, Deep Q-Networks (DQN) (Mnih et al., 2015), stores agent-environment interactions as tuples in a fixed-capacity buffer and randomly samples small batches during training. However, it treats all experiences equally, failing to prioritize important ones. Advancements in experience replay can be categorized into three categories. i) Experience Priority Weighting (Mei et al., 2023; Oh et al., 2022; Lahire et al., 2022; Horgan et al., 2018): These approaches prioritize experiences based on relevance, often using metrics like Temporal Difference (TD) errors. A notable example is Prioritized Experience Replay (PER) (Schaul et al., 2016), which employs non-uniform sampling to emphasize important experiences. However, PER faces challenges such as gradient estimation bias and delayed priority updates for infrequent experiences, exacerbating the issue of forgetting. ii) Noise Detection and Filtering (Yu et al., 2024; Vezhnevets et al., 2017; Gu et al., 2017): These approaches aim to improve the quality of replayed data by identifying and removing noisy experiences. For instance, Zhang and Sutton (2017) introduced a technique to filter noisy experiences, enhancing learning. Despite their benefits, such approaches may inadvertently discard important experiences in dynamic environments and require careful tuning to avoid overfitting to noise. iii) Memory Augmentation and Multi-Network Strategies (Lu et al., 2023; Yang et al., 2022; Buzzega et al., 2020): These approaches utilize auxiliary networks to store and manage historical experiences, ensuring that important experiences are retained while less relevant ones are down-weighted. A prominent work is Topological Experience Replay (TER) (Hong et al., 2022), which organizes experiences as a graph and updates it using reverse value backup with Breadth-First Search (BFS). While TER organizes experience replay, it incurs high computational costs and may be unstable with noisy or rapidly changing

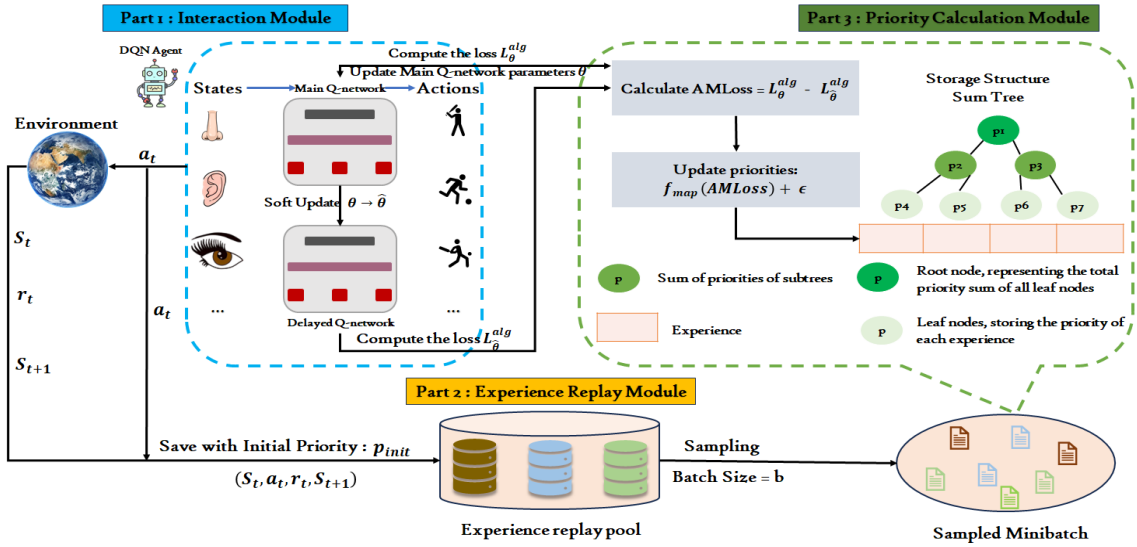


Figure 2: Description of the AMLoss method modules.

policies.

In summary, despite recent advances, no existing method fully addresses the priority distortion issues stemming from forgetting and noisy experiences. In contrast, our AMLoss improves experience replay by dynamically adjusting priorities based on the loss difference between the Main and Delayed Q-networks. This mechanism mitigates the impact of delayed priority updates—thereby reducing experience forgetting—and enhances sensitivity to noisy samples. Moreover, it is compatible with existing PER parameter designs, easy to implement, and adds minimal computational overhead, resulting in superior robustness against forgetting and noisy experiences.

3 Priority Experience Replay Based on Adjustable Mirror Loss (AMLoss)

As shown in Figure 2, the AMLoss method consists of three modules: 1) *Interaction module*, which facilitates the interaction between the DQN agent and the environment. 2) *Experience replay module*, which stores and samples the experience generated from the interaction. 3) *Priority calculation module*, which evaluates experience priorities based on AMLoss and samples them according to their priority using the Sum Tree structure, which facilitates efficient hierarchical storage and weighted sampling. It enables optimized experience replay and priority adjustment. By prioritizing learning experiences that contribute more to improvement, AMLoss mitigates the effects of experience forgetting and reduces the impact of noisy experiences.

3.1 Part 1 : Interaction Module

The interaction module generates experience data through the interaction between the RL agent[1] and the environment. The DQN agent contains two neural networks. The Main Q-network estimates the Q-values for each action, while the Delayed Q-network computes the target Q-values to stabilize learning. The DQN agent selects the action $\boldsymbol{a_t}$ with the highest Q-value using the $\boldsymbol{\epsilon}$-greedy strategy and receives feedback from the environment, including the current reward $\boldsymbol{r_t}$ and the next state $\boldsymbol{s_{t+1}}$.

The parameters of the Main Q-network $\boldsymbol{\theta}$ are updated through training, while the parameters of the Delayed Q-network $\hat{\boldsymbol{\theta}}$ are synchronized gradually using a soft update mechanism:

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

where $\boldsymbol{\tau} \in (\mathbf{0}, \mathbf{1}]$ is the soft update coefficient.

The agent computes the loss $\boldsymbol{L_\theta^{alg}}$ on the Main Q-network and $\boldsymbol{L_{\hat{\theta}}^{alg}}$ on the Delayed Q-network, which will be used for the subsequent calculation of AMLoss and priority adjustment (Eq. 3).

3.2 Part 2 : Experience Replay Module

The experience replay module is responsible for storing and sampling the experience data generated by the interaction module based on the AMLoss priority calculated from the third module. Each interaction generates an experience tuple $\langle \boldsymbol{s_t}, \boldsymbol{a_t}, \boldsymbol{r_t}, \boldsymbol{s_{t+1}} \rangle$, which is stored in the experience

[1] Since DQN is a commonly used baseline for task-oriented dialogue policy, we use DQN for method description and validation in this paper.

pool and assigned an initial priority $\boldsymbol{P_{init}}$. During the training phase, the experience pool is sampled based on AMLoss priority, and a batch of experience (with batch size $\boldsymbol{b}$) is selected for training. For the sampled experience, a hierarchical storage structure (Sum Tree) is used to manage the priorities, enabling priority-based weighted sampling (Li et al., 2022; Emmons et al., 2020).

3.3 Part 3 : Priority Calculation Module

The priority calculation module is the core of this method. It focuses on optimizing the experience replay process by combining the Sum Tree structure and the AMLoss priority update method.

The Sum Tree is a binary tree data structure specifically designed to store and manage experience priorities. Its characteristic is that each node stores the sum of its children's priorities, which allows weighted sampling and priority updates to be performed in $\mathcal{O}(\log n)$ time complexity:

- Leaf nodes: store the priority value $\boldsymbol{p}_i$ of each experience.
- Intermediate nodes: store the sum of the priorities of all their child nodes.
- Root node: stores the sum of all leaf node priorities, representing the total priority of the entire experience pool $\sum_{i=1}^{n} \boldsymbol{p}_i$.

With this structure, a weighted sampling mechanism can efficiently sample experience from the experience pool. The specific operation is as follows: A random value $\boldsymbol{u}$ is generated within the range $[0, \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{p}_i]$. Starting from the root node, we recursively compare $\boldsymbol{u}$ with the priority values of the left and right child nodes, and eventually locate the corresponding leaf node (i.e., the corresponding experience tuple). It ensures that the probability $\boldsymbol{P}(\boldsymbol{i})$ of sampling an experience with higher priority is proportional to its priority $\boldsymbol{p}_i$, i.e.,

$$\boldsymbol{P}(\boldsymbol{i}) = \frac{\boldsymbol{p}_i}{\sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{p}_j} \tag{2}$$

To dynamically adjust the priority of experience, we calculate the AMLoss priority of each experience in the priority calculation module. The process consists of the following steps:

1) AMLoss Calculation: AMLoss is the core metric for measuring the priority of experience. It is defined by the difference in losses between the Main Q-network and the Delayed Q-network, and can be computed as:

$$\boldsymbol{AMLoss}_i = \boldsymbol{L}_{\boldsymbol{\theta}}^{\boldsymbol{alg}}(i) - \boldsymbol{L}_{\hat{\boldsymbol{\theta}}}^{\boldsymbol{alg}}(i) \tag{3}$$

Where: $\boldsymbol{L}_{\boldsymbol{\theta}}^{\boldsymbol{alg}}(i)$ and $\boldsymbol{L}_{\hat{\boldsymbol{\theta}}}^{\boldsymbol{alg}}(i)$ are the network loss values for the experience $\boldsymbol{i}$ in the Main Q-network and the Delayed Q-network, respectively. AMLoss measures the learnability of the experience by calculating the loss difference between the Main and Delayed Q-networks for a given experience sample, providing a reliable metric for prioritizing experience selection.[2]

2) Priority Update: Based on the value of AMLoss, we update the experience priority using a mapping function $\boldsymbol{f_{map}}$. The mapping function can be nonlinear (e.g., exponential or smooth function), and the specific form is as follows:

$$\boldsymbol{p}_i = \boldsymbol{f_{map}}(\boldsymbol{AMLoss}_i) + \boldsymbol{\epsilon} \tag{4}$$

Where $\boldsymbol{f_{map}}$ maps AMLoss to positive values to ensure the priority remains positive. For example, $\boldsymbol{f_{map}} = \max(0, \boldsymbol{AMLoss}_i)$. $\boldsymbol{\epsilon}$ is a small positive value (such as 10^{-6}), which ensures that all experiences have a non-zero sampling probability, preventing sampling dead zones (Lee et al., 2019).

The updated priorities are stored in the leaf nodes of the Sum Tree and recursively update the priority sums of the intermediate nodes and root node to maintain the consistency of the entire structure.

3.4 Implementation

The general process of the AMLoss-based DP method is as follows: The experience replay pool and the Delayed Q-network are initialized. During interaction with the environment, the current state is observed, an action is selected and executed, and the reward and next state are recorded, along with the transition $\{\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{r}_t, \boldsymbol{s}_{t+1}\}$ stored in the replay pool with an initial priority. In each iteration, a small batch of experiences is sampled and AMLoss is calculated by its loss in the Main and the Delayed Q-network, and experience priorities are adjusted. Finally, the Delayed Q-network is updated according to the Offline RL algorithm. For more details about the algorithm, please refer to the Appendix A.1.

4 Experiment

In this section, we assess the effectiveness of AMLoss on four commonly used, publicly available

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

TOD datasets: movie-ticket booking, restaurant reservation, taxi booking, and MultiWOZ 2.1. The objectives of this experiment are:[3]

I) Assess the effectiveness of AMLoss compared to baseline methods: Examine the advantages of AMLoss in handling small amounts of noisy experiences and compare its performance with related baseline methods.

II) Examine the performance of AMLoss in addressing the issue of noisy experiences: Investigate how AMLoss effectively handles strong noise interference in the data and demonstrate its stability in noisy environments by visualizing the trend of experience priorities.

III) Investigate AMLoss's ability to address the issue of forgetting experiences: Evaluate how AMLoss retains important experience during long-term training and prevents forgetting, showcasing its advantages by visualizing the trend of experience priorities.

IV) Evaluate AMLoss through human assessment: Analyze the performance of AMLoss in real-world dialogue scenarios through human evaluation of TODs' outputs.

4.1 Baselines

We compare the AMLoss method with several baselines: Deep Q-network (DQN) with Random Experience Replay (RER) (Mnih et al., 2015), which randomly samples state-transition tuples from the replay buffer without prioritization. Prioritized Experience Replay (PER) (Schaul et al., 2016), which prioritizes experiences based on high TD errors to enhance learning efficiency. Topology Experience Replay (TER) (Hong et al., 2022), which organizes the agent's experiences into a graph, where each edge tracks dependencies between states and value backups are performed via breadth-first search from terminal states.

4.2 Experimental Settings

4.2.1 Datasets

This study uses four datasets, including both single-domain and multi-domain datasets, that are widely used in TODs research: MultiWOZ 2.1 (Budzianowski et al., 2018) and Microsoft Dialogue Challenge (1-3) (Li et al., 2018). The domain and feature information of different datasets are shown in Table 1. For more information about the datasets, please refer to the Appendix A.2.

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

Dataset	Domain	Scale
MultiWOZ 2.1	Attraction Hospital Police Hotel Restaurant Taxi Train	Dialogue scale: 8,438 Dialogue rounds: 115,424 Average number of conversation rounds: 13.68 Number of slots: 25
Microsoft Dialogue Challenge 1	Movie	Dialogue scale: 2890; Intention: 11; Slot: 29
Microsoft Dialogue Challenge 2	Restaurant	Dialogue scale: 4103; Intention: 11; Slot: 30
Microsoft Dialogue Challenge 3	Taxi	Dialogue scale: 3094; Intention: 11; Slot: 29

Table 1: Dataset statistics for various dialogue tasks.

4.2.2 Implementation Details

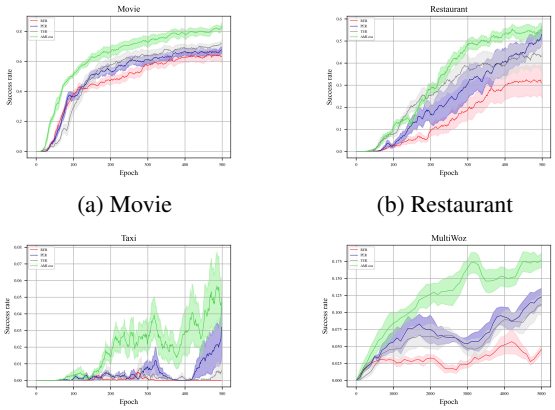
The probability of sampling data points is related to the priority through Eq. 4, ensuring that the priority remains non-negative. Since the Q-value method uses Mean Squared Error (MSE) loss, the priority is inherently non-negative (Mnih et al., 2015). In contrast, AMLoss computes the difference in MSE loss, which does not guarantee the same property. When the delayed network is updated along with the main network, this value can become zero. However, after a single update, it quickly becomes non-zero. Therefore, we need to create a mapping function f_{map} for the AMLoss error that is monotonically increasing and non-negative for all values.

In practice, we found that clipping negative values to zero and adding a small value to ensure that the samples have a minimum probability works effectively, defined as $\max(0, \text{AMLoss}) + \epsilon$, where ϵ is a small positive constant, ensuring that all samples have a non-zero priority.

4.3 Main Result

As shown in Figure 3, the results demonstrate that under the 10% noise condition, AMLoss significantly outperforms the fundamental methods (RER, PER, TER) across four datasets, achieving higher success rates and faster convergence. In particular, on the Movie and Restaurant datasets, AMLoss shows clear advantages, reaching higher success rates earlier and maintaining stability in the later stages. In the Taxi and MultiWoz 2.1 datasets, although the complexity of the task and noise impact cause all methods to exhibit lower success rates with greater fluctuations, AMLoss still shows stronger noise resilience in the later stages and eventually outperforms the other methods. Overall, the AMLoss method significantly improves the robustness and success rate of TODs under low noise conditions, especially demonstrating excellent noise resilience in more complex tasks.

[4]Through statistical analysis, we observed that noise in real-world applications typically ranges from 8% to 12%. To



(a) Movie (b) Restaurant

(c) Taxi (d) MultiWoz 2.1

Figure 3: Performance comparison of different methods for TODs across four datasets in a normal environment (10% noise).[4]The subfigures show the average dialogue success rate over epochs, with shaded areas representing the error bounds. The MultiWOZ results are based on the average of five experiments, while the others are based on the average of ten experiments.

4.4 Validation of noisy experience

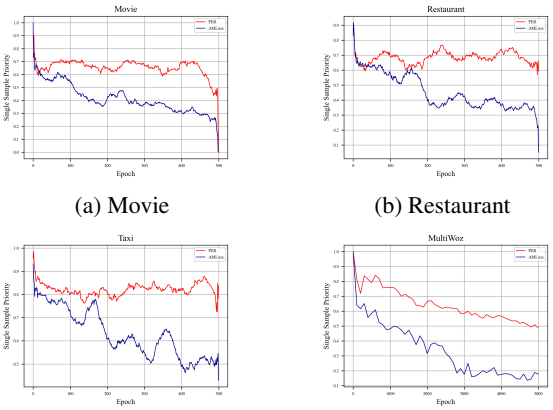
4.4.1 Verification of Experimental Results

In this experiment, to verify the effectiveness of the AMLoss method in identifying and avoiding noisy experiences, we simulated different levels of noisy data by adding slot noise and observed the performance changes of AMLoss and baseline methods as the probability of noisy experiences increased.

As shown in Figures 4 and 5, AMLoss outperforms all other methods in all scenarios, demonstrating higher success rates and better stability, especially in noisy environments. It highlights its ability to identify and avoid noisy experiences effectively. Although PER performs better than RER, its performance still lags behind AMLoss, especially in noisy conditions. The increased probability of noisy experiences in such environments leads PER to repeatedly prioritize this experience, which is confusing training. In contrast, AMLoss adjusts the priority of noisy experience by comparing losses between the Main and Delayed 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 (Zhao et al., 2020).

4.4.2 Visual Analysis of noisy experience

Noisy experience refers to certain sample data that contain intrinsic uncertainty or randomness, which causes the priority calculation of this experience to deviate from its actual learning value. In such cases, the traditional PER method, which relies solely on TD errors to calculate priorities, is easily affected by noise, leading to an overestimation of the priority of experience.



(a) Movie (b) Restaurant

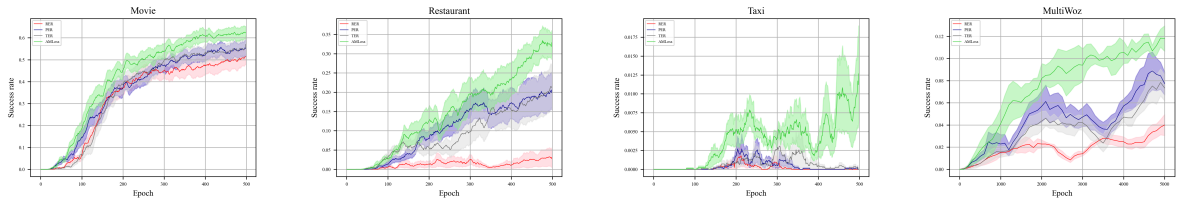
(c) Taxi (d) MultiWoz 2.1

Figure 6: Sample Priority Curve Based on PER and AMLoss Methods. This figure compares the changes in sample priority based on the PER and AMLoss methods across four datasets, highlighting AMLoss's effectiveness in handling noisy experience.

To analyze the effectiveness of the AMLoss method in handling noisy experiences, we dynamically track the changes in experience priority over time. In the experiment, we select data samples containing noise and record their priority distributions at different training stages (e.g., different epochs). By comparing with the PER method, we observe whether AMLoss can more effectively suppress noise interference, resulting in more stable experience priorities closer to the true learning value.

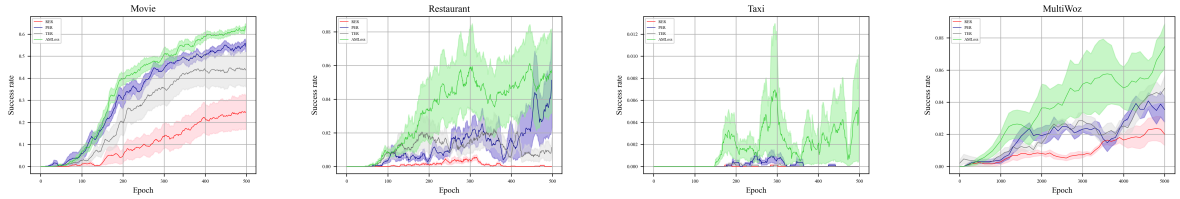
As shown in Figure 6, when an experience contains noise, the AMLoss method effectively reduces the priority of noisy experiences. As training progresses, the priority of individual experiences under AMLoss gradually decreases, reflecting its ability to eliminate noisy experiences. In contrast, under the PER method, the priority of noisy experiences remains high and is difficult to remove even in the later stages of training. It causes the model to continually rely on high-noise experiences, which is detrimental to the learning process. It indicates that

simulate this, we set the noise level at 10% in our experiments (He and McAuley, 2016). This models common errors in dialogue systems without compromising performance. The 10% noise setting allows us to evaluate baseline methods' effectiveness in noisy environments and provides a controlled framework for assessing their robustness and reliability.



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

Figure 4: Performance comparison of different methods for TODs across four datasets in a noise-enhanced environment (15% noise).



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

Figure 5: Performance comparison of different methods for TODs across four datasets in a noise-enhanced environment (20% noise).

AMLoss demonstrates a clear advantage in handling noisy experiences, significantly improving the efficiency of experience replay.

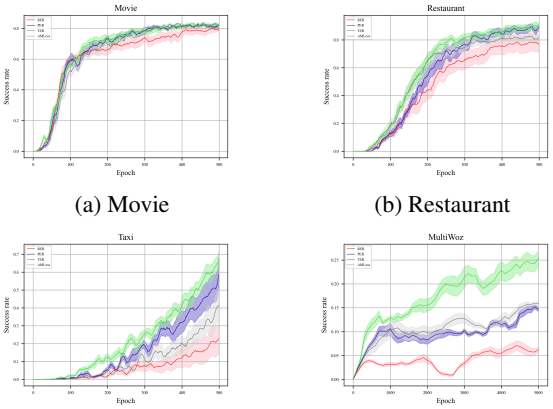
4.5 Validation of Forgetting Experience

4.5.1 Verification of Experimental Results

In this experiment, we constructed a noise-free environment to evaluate the performance of the AMLoss method, aiming to verify that even when noise is eliminated, AMLoss can still perform excellently, particularly in more complex tasks with larger state spaces in TODs.

As illustrated in Figure 7, even after eliminating the impact of noisy experiences, the AMLoss method still exhibits strong performance, especially notable improvements in the complex TAXI domain and the MultiWOZ 2.1 multi-domain task. This is because, in RL, the model tends to forget earlier experiences as training progresses, resulting in the loss of important reference information for decision-making. In complex tasks, where state and action spaces are vast, it becomes increasingly challenging for the model to retain crucial experience. Without a mechanism to preserve key experience, the model's learning process is susceptible to "forgetting," which hampers decision-making accuracy.

AMLoss method addresses this issue by reusing forgetting experiences, thus mitigating the detrimental effects of forgetting during training.



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

Figure 7: Performance comparison of different methods for TODs across four datasets in a noise-free environment.

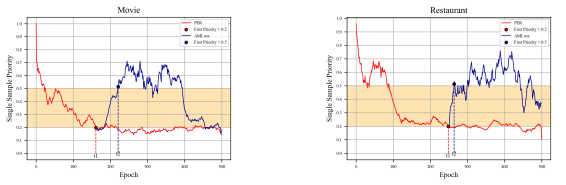
4.5.2 Visual Analysis of Forgetting Experience

Forgetting experience refers to the loss of valuable early experiences during long-term training, particularly in large-scale and complex task scenarios, where continuous updates to model parameters lead to the gradual forgetting of certain experience (Lahire et al., 2022; van Hasselt et al., 2016; Schaul et al., 2016). As a result, its priority continues to decrease, and useful experiences may fail to be revisited. This issue is especially prominent in the PER method, where the mechanism does not effectively balance the importance of long-term and short-term learning, causing early experiences to be forgotten by the model.

To demonstrate the effectiveness of the AMLoss in reusing forgetting experiences, we plan to: **1)**

Normalize and store the priority of experience in both PER and AMLoss. **2)** Use the priority to classify experience as either forgetting or reused, where a priority below A indicates the release of an experience, and a priority above B indicates its reuse.

The relationship between priority and sampling probability is shown in Eq. 4, which means that a lower priority corresponds to a smaller sampling probability. We set (A, B) = (0.2, 0.5) as the priority threshold for forgetting and reusing experience. The figure shows that the priority of PER drops below the threshold (A = 0.2) at t1, indicating that some low-priority experience is gradually forgotten. In contrast, the priority of AMLoss rises and exceeds the threshold (B = 0.5) at t2. AMLoss efficiently uses experience replay by reusing forgetting experiences and dynamically adjusting key experiences, thereby improving learning efficiency and model convergence performance.



(a) Movie (b) Restaurant

(c) Taxi (d) MultiWoz 2.1

Figure 8: Sample Priority Curve Based on PER and AMLoss Methods. Compared with Figure 6, a priority boundary for determining forgetting and learning is introduced to describe the process of reusing forgetting experience in the AMLoss method.

4.6 Human Evaluation

In this experiment, we conducted a human evaluation to assess the quality of the model output. Human evaluation is essential to measure the performance of the model in terms of accuracy, fluency, etc., because automatic evaluation metrics may not fully capture these characteristics (Braggaar et al., 2023). The evaluation criteria include fluency, i.e., the naturalness of the output text, and consistency, i.e., the logical consistency of the output. A total of 20 participants were included, including experts in the field and ordinary users. The expert group members had certain task background knowledge, while the ordinary users represented a wider user group. During the evaluation process, each participant talked to the trained model and scored each model according to the above criteria. Accuracy was scored as 1, and no accuracy was scored as 0. Fluency was scored from 1 to 5, from low to high. As shown in Table 2, the results are consistent with the simulated experiments, where our AMLoss achieved the best performance.

Model	Domain	Accuracy	Fluency
RER	Movie	70%	3.2
	Restaurant	65%	2.55
	Taxi	80%	3.1
PER	Movie	65%	2.8
	Restaurant	75%	2.65
	Taxi	50%	1.95
TER	Movie	70%	2.75
	Restaurant	75%	3.3
	Taxi	65%	2.85
AMLoss	Movie	85%	3.8
	Restaurant	80%	3.5
	Taxi	90%	3.35

Table 2: Human evaluation of different models. For additional human evaluation results under noisy conditions, please refer to the Appendix A.3.

5 Conclusion

This paper proposes a novel Adjustable Mirror Loss (AMLoss) method to address the priority distortion problem caused by forgetting and noisy experiences in Offline RL-based dialogue policies. By comparing the loss differences between the Main Q-network and the Delayed Q-network, AMLoss evaluates the incremental contribution of experiences to dialogue policy improvement, optimizing priority calculation and mitigating distortion. Using the Delayed Q-network to represent experience retention, AMLoss avoids simple loss-based sampling, as higher losses may arise from noisy experiences while still revisiting valuable experiences to address forgetting. We validate this method on four TOD datasets with varying noise levels and analyze its effectiveness in handling forgetting and noisy experiences. It is noted that AMLoss is compatible with PER, making its implementation straightforward—experience priority is assessed by calculating the Delayed Q-network loss based on $\mathbf{L}^{alg}$ and $\hat{\theta}$. Thus, our AMLoss method aligns with PER and can be applied to classical Offline RL to improve performance further.

Limitations

Even though AMLoss shows strong effectiveness in advancing prioritized experience replay, our work has some limitations. AMLoss, due to its design for priority calculation, is primarily focused on optimizing experience replay in Offline RL. In offline 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 AMLoss. As a result, AMLoss is best suited for Online RL and may not be ideal for scenarios involving Online RL.

References

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

Anouck Braggaar, Christine Liebrecht, Emiel van Miltenburg, and Emiel J. Krahmer. 2023. Evaluating task-oriented dialogue systems: A systematic review of measures, constructs and their operationalisations. *CoRR*, abs/2312.13871.

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.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3):39:1–39:45.

Scott Emmons, Ajay Jain, Michael Laskin, Thanard Kurutach, Pieter Abbeel, and Deepak Pathak. 2020. Sparse graphical memory for robust planning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Giorgio Franceschelli and Mirco Musolesi. 2024. Reinforcement learning for generative AI: state of the art, opportunities and open research challenges. *J. Artif. Intell. Res.*, 79:417–446.

Mingqi Gao, Xinyu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2024. Llm-based NLG evaluation: Current status and challenges. *CoRR*, abs/2402.01383.

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.

Hossein Hassani, Soodeh Nikan, and Abdallah Shami. 2025. Improved exploration-exploitation trade-off through adaptive prioritized experience replay. *Neurocomputing*, 614:128836.

Ruining He and Julian J. McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 507–517. ACM.

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.

Niel Teng Hu, Xinyu Hu, Rosanne Liu, Sara Hooker, and Jason Yosinski. 2021. When does loss-based prioritization fail? *CoRR*, abs/2107.07741.

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.

Thibault Lahire, Matthieu Geist, and Emmanuel Rachelson. 2022. Large batch experience replay. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 11790–11813. PMLR.

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.

Mengyuan Li, Arman Kazemi, Ann Franchesca Laguna, and X. Sharon Hu. 2022. Associative memory based experience replay for deep reinforcement learning. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2022, San Diego, California, USA, 30 October 2022 - 3 November 2022*, pages 135:1–135:9. ACM.

Qiyang Li, Aviral Kumar, Ilya Kostrikov, and Sergey Levine. 2023. Efficient deep reinforcement learning requires regulating overfitting. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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.

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*.

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.

Steven D. Morad, Ryan Kortvelesy, Stephan Liwicki, and Amanda Prorok. 2023. Reinforcement learning with fast and forgetful memory. 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*.

Daniel Eugênio Neves, Lucila Ishitani, and Zenilton Kleber Gonçalves do Patrocínio Jr. 2025. Advances and challenges in learning from experience replay. *Artif. Intell. Rev.*, 58(2):54.

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.

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*.

Hado van Hasselt, Arthur Guez, and David Silver. 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.

Che Wang and Keith W. Ross. 2019. Boosting soft actor-critic: Emphasizing recent experience without forgetting the past. *CoRR*, abs/1906.04009.

Hongru Wang, Lingzhi Wang, Yiming Du, Liang Chen, Jingyan Zhou, Yufei Wang, and Kam-Fai Wong. 2023. A survey of the evolution of language model-based dialogue systems. *CoRR*, abs/2311.16789.

Heng-Da Xu, Xian-Ling Mao, Puhai Yang, Fanshu Sun, and Heyan Huang. 2024. Rethinking task-oriented dialogue systems: From complex modularity to zero-shot autonomous agent. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 2748–2763. Association for Computational Linguistics.

Biao Yang, Fucheng Fan, Rongrong Ni, Jie Li, Chu Kiong Loo, and Xiaofeng Liu. 2022. Continual learning-based trajectory prediction with memory augmented networks. *Knowl. Based Syst.*, 258:110022.

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

Shangtong Zhang and Richard S. Sutton. 2017. A deeper look at experience replay. *CoRR*, abs/1712.01275.

Yangyang Zhao, Zhenyu Wang, Kai Yin, Rui Zhang, Zhenhua Huang, and Pei Wang. 2020. Dynamic reward-based dueling deep dyna-q: Robust policy learning in noisy environments. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of*

Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 9676–9684. AAAI Press.

A Appendix

A.1 Algorithm of AMLoss

Algorithm 1:
Computing AMLoss for Prioritization

Require:

- Off-policy RL algorithm A with loss function L^{alg}
- Main Q-network parameters $\boldsymbol{\theta}$
- Delayed Q-network parameters $\hat{\boldsymbol{\theta}}$
- Experience replay pool B
- Initial priority P_{init}
- AMLoss Normalization Function f_{map}
- Epsilon priority $\boldsymbol{\epsilon}$
- Dialogue training timesteps T
- Iteration steps per training timesteps T_{iter}
- Sampling batch size b

1: Initialize Experience replay pool B as empty;
2: Initialize Delayed 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 priority P_{init};

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

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

4: Update Delayed Q-network following RL algorithm A;

A.2 Datasets of TODs

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 AMLoss method in multi-domain, multi-task environments, particularly in handling noisy experiences, forgetting experiences, and human evaluation performance.

A.3 Human Evaluation for Models

Model	Domain	Accuracy	Fluency
RER+10%Noise	Movie	60%	2.6
	Restaurant	65%	2.05
	Taxi	60%	2.1
RER+15%Noise	Movie	65%	2.15
	Restaurant	65%	2.45
	Taxi	55%	2
RER+20%Noise	Movie	75%	2.4
	Restaurant	55%	2.45
	Taxi	65%	2
PER+10%Noise	Movie	55%	1.15
	Restaurant	70%	1.25
	Taxi	60%	1.9
PER+15%Noise	Movie	70%	3
	Restaurant	75%	2.65
	Taxi	80%	2.85
PER+20%Noise	Movie	70%	3.15
	Restaurant	65%	2.8
	Taxi	70%	3
TER+10%Noise	Movie	60%	2.25
	Restaurant	55%	1.75
	Taxi	65%	2.05
TER+15%Noise	Movie	50%	1.95
	Restaurant	60%	2.5
	Taxi	55%	2.25
TER+20%Noise	Movie	45%	2.2
	Restaurant	50%	2.55
	Taxi	35%	1.4
AMLoss+10%Noise	Movie	75%	3.65
	Restaurant	85%	3.05
	Taxi	70%	3.5
AMLoss+15%Noise	Movie	80%	3.7
	Restaurant	85%	4.15
	Taxi	70%	3
AMLoss+20%Noise	Movie	75%	3.1
	Restaurant	65%	2.85
	Taxi	75%	3.5

Table 3: Human evaluation results of the models under different noisy conditions.