

Anti-Overestimation Dialogue Policy Learning for Task-Completion Dialogue System

Chang Tian[†] and Wenpeng Yin* and Marie-Francine Moens[†]

[†]Department of Computer Science, KU Leuven

*LanguageX Lab, Temple University

chang.tian@kuleuven.be

Abstract

A dialogue policy module is an essential part of task-completion dialogue systems. Recently, increasing interest has focused on reinforcement learning (RL)-based dialogue policy. Its favorable performance and wise action decisions rely on an accurate estimation of action values. The overestimation problem is a widely known issue of RL since its estimate of the maximum action value is larger than the ground truth, which results in an unstable learning process and suboptimal policy. This problem is detrimental to RL-based dialogue policy learning. To mitigate this problem, this paper proposes a dynamic partial average estimator (DPAV) of the ground truth maximum action value. DPAV calculates the partial average between the predicted *maximum* action value and *minimum* action value, where the weights are dynamically adaptive and problem-dependent. We incorporate DPAV into a deep Q-network as the dialogue policy and. Our method can achieve better or comparable results compared to top baselines on three dialogue datasets of different domains *with a lower computational load*. In addition, we also theoretically prove the convergence and derive the upper and lower bounds of the bias compared with those of other methods.

1 Introduction

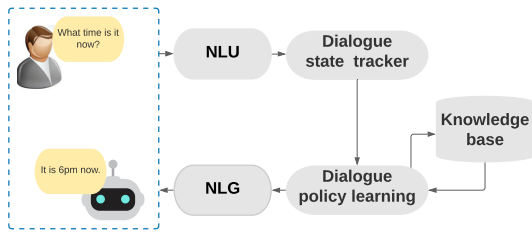
Task-completion dialogue systems are commonly implemented in two schemes. One is by end-to-end training, such as (Zhang et al., 2020a). The other is a pipeline framework (Chen et al., 2017), which typically consists of four modules that are independently trained, as shown in Figure 1a: natural language understanding (NLU), dialogue state tracker (DST), dialogue policy learning (DPL) and natural language generation (NLG). For this pipeline-style dialogue system, the conversation text from a user is first fed to the NLU module, where the user utterance is parsed into semantic slots for DST. DST manages the inputs of each dialogue turn together

with the dialogue history. Then DST outputs the current dialogue state embedding to the DPL module, where a dialogue action is taken based on current dialogue state and knowledge base data. The NLG module maps the selected dialogue action into natural language to converse with the user.

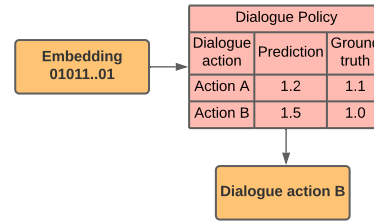
Reinforcement learning (RL) algorithms, specifically Q-learning (Watkins and Dayan, 1992) based algorithms, have become a mainstream method for training the dialogue policy module (Peng et al., 2018; Zhang et al., 2020b). For each step, the policy agent updates its action value ¹ estimate as the sum of the observed reward and the estimated maximal action value in the next state. However, this update rule suffers from an overestimation problem (Hasselt, 2010): mostly the estimated maximal action value is larger than the ground truth. The overestimation problem causes that the dialogue policy module has inaccurate action values estimations after the training, which misleads the dialogue policy to choose the wrong dialogue action (see the wrong dialogue action in Figure 2). Some prior studies have tried to address this problem in domains like video game playing and multi-agent systems, but they either suffered from the underestimation problem (Hasselt, 2010; Lan et al., 2020) or required heavy computational load, such as those ensemble methods (Anschel et al., 2017; Lan et al., 2020; Lee et al., 2021).

In this work, we propose dynamic partial average (DPAV), a novel approach to mitigate the overestimation problem specifically for the task-completion dialogue policy. DPAV utilizes the *partial average* between the predicted *maximal* action value and the predicted *minimal* action value to estimate the ground truth maximum action value, where the weights are dynamically adaptive and problem-dependent. The rationale here is that

¹This value is the expected return for taking the action under a certain state, and it is represented as the Q value of Q-learning.



(a) Four modules in the pipeline framework



(b) The overestimation problem in RL-based dialogue policy

Figure 1: Task-completion dialogue system

Overestimation version:	Anti-overestimation version:
wrong dialogue action within conversation	correct dialogue action within conversation
Turn 0 usr: Can I get some tickets for room?	Turn 0 usr: Can I get some tickets for room?
Turn 1 sys: How many tickets do you need?	Turn 1 sys: How many tickets do you need?
Turn 2 usr: I want 2 tickets please!	Turn 2 usr: I want 2 tickets please!
Turn 3 sys: 9:30 pm is available.	Turn 3 sys: 9:30 pm is available.
Turn 4 usr: I want to watch at seattle.	Turn 4 usr: I want to watch at seattle.
Turn 5 sys: How many tickets do you need?	Turn 5 sys: amc pacific place 11 theater is available.
Dialogue action: request_numberofpeople	Correct dialogue action: inform_theater
Turn 6 usr: I want 2 tickets please!	Turn 6 usr: I want to set it up tomorrow.
Turn 7 sys: 9:30 pm is available.	Turn 7 sys: Great - I was able to purchase 2 tickets for you to see room tomorrow at amc pacific place 11 theater in seattle at 9:30 pm.
Turn 8 usr: I want to watch at seattle.	Turn 8 usr: Thank you!
Turn 9 sys: How many tickets do you need?	Turn 9 sys: Thank you!
Dialogue action: request_numberofpeople	Simulation episode : Success
Turn 10 usr:Turn 11 sys:Turn.....	
Simulation episode : Failure	

Figure 2: Wrong and correct dialogue actions

DPAV learns the optimal trade-off between the predicted maximal action value and the predicted minimal action value so that the dialogue policy learning procedure will be more reasonable and stable. Our system not only yields a better dialogue process (see Figure 2), but also has much lower computational cost compared to ensemble models.

Overall, our main contributions are as follows: (i) This is the first work to investigate and handle the overestimation problem of the reinforcement learning framework for task-completion dialogue systems. (ii) We propose a novel and effective approach, the dynamic partial average DPAV, which can alleviate the overestimation problem with lower computational load. (iii) We theoretically prove the convergence and derive the upper and lower bounds of our method to claim its effectiveness.

2 Related Work

Dialogue Policy. The dialogue policy module makes a dialogue decision given the current state (Zhang et al., 2019). Early methods are rule-based (Chen et al., 2017). Since handcrafted rules are non-extensible and resource-consuming (Zhao et al., 2021), deep reinforcement learning (DRL) has become a mainstream method for training dialogue policies (Wu et al., 2019; Wang et al., 2020; Zhao et al., 2021). Task-completion dialogue policy learning is often regarded as an RL problem (Zhang et al., 2021).

Overestimation Bias. The value-based algorithm Q-learning, a common unit of the dialogue policy module, suffers the overestimation bias (Thrun and Schwartz, 1993; Hasselt, 2010). Prior studies addressed the problem in multiple ways, including (1) bias compensation with additive pseudo costs and (2) a variety of estimators. Bias-corrected Q-Learning (Lee et al., 2013) subtracts a quantity from the target but this method cannot address the bias from the function approximation (Pentaliotis and Wiering, 2021). It is known that the bias compensation method is labor involved and time consuming (Anwar and Patnaik, 2008; Lee and Powell, 2012). Double Q-learning (Hasselt, 2010) trades overestimation bias for underestimation bias using the double estimator. Since underestimation bias is not preferable (Hasselt, 2010; Lan et al., 2020), Weighted Q-learning proposes (D’Eramo et al., 2016; Zhang et al., 2017) the weighted estimator for the maximal action value based on a weighted average of estimated actions values. However, the weights computation is only practical in a tabular setting (D’Eramo et al., 2017). Our work differs from the foregoing in that it proposes a new estimator which could be generalized into the deep Q-learning network setting.

Overestimation bias is more problematic in the deep Q-learning network (DQN) algorithm (Fan et al., 2020) due to the function approximation errors of DRL. Polishing estimation tricks of a single model and using ensemble models are two mainstream solutions. Double Q-learning is subsequently adapted to a neural network as Double DQN (Van Hasselt et al., 2016), and Duel DQN proposes a new action value estimation scheme (Wang et al., 2016). But the two methods still suffer the bias of double estimator and maximum estimator, respectively. Another approach against overestimation bias is based on the idea of ensembling. Averaged DQN controls the estimation bias by tak-

ing the average over action values of multiple target networks (Ansel et al., 2017). Later, (Lan et al., 2020) claims that an average operation will never completely remove the overestimation bias, and they propose the Maxmin DQN which takes a minimum from multiple maximums of different ensemble units to estimate the maximum action value in a selective process. Then, (Kuznetsov et al., 2020) recognizes that Maxmin DQN also suffers underestimation bias and that the bias control is coarse. Recently, the SUNRISE method uses the uncertainty estimates of the ensemble. But it only down-weights the biased estimation (Lee et al., 2021). In this work, the model only uses a value function instead of a combination of multiple value functions and tailors the predicted maximum and minimum of a value function to approximate the optimal action value. Our work does not move towards underestimation and avoid the computational complexity of ensemble models.

3 Preliminary

3.1 Problem Definition

Even though an unbiased estimator does not exist (D’Eramo et al., 2016), the maximum estimator (Watkins and Dayan, 1992) and double estimator (Hasselt, 2010; Van Hasselt et al., 2016) are the most representative among the relevant works.

Maximum estimator (ME). This method is used by deep Q-learning to approximate the ground truth maximum action value of the following state by maximizing over a set of action values $Q(s_{t+1}, \cdot)$. It represents the target y^{DQN} for taking a possible action a under the state s_{t+1} as:

$$y^{DQN} = r_{t+1} + \gamma \max_a Q(s_{t+1}, a; \theta^-) \quad (1)$$

where r_{t+1} is the reward, γ is the discount value for future rewards and θ^- is parameters of the target network. As Smith and Winkler (2006) found, the estimate of ME is larger than the ground truth (i.e., the estimated maximum value of the following state, $\max_a Q(s_{t+1}, a; \theta^-)$ is overestimated), which results in the biased loss:

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

where m is the RL experience replay pool and θ is parameters of the DQN model. Thus, the $Q(s_t, \cdot)$ will not be perfectly accurate after training,

Double estimator (DE). This method (Hasselt, 2010; Van Hasselt et al., 2016) is used by deep Q-learning to solve the overestimation problem of ME in DQN. The Double DQN has two estimators, and one estimator decides the action index while the other estimator evaluates the action value of the selected action. Then Double DQN (DDQN) uses the evaluated action value to estimate the ground truth maximum action value of state s_{t+1} :

$$y^{DDQN} = r_{t+1} + \gamma Q(s_{t+1}, \arg \max_a Q(s_{t+1}, a; \theta^+); \theta^-). \quad (3)$$

However, DE suffers from the underestimation problem and does not guarantee better estimation than ME (Lan et al., 2020).

3.2 Problem in Dialogue Policy

Q-learning is a common unit of RL-based dialogue policies. The overestimation bias of ME propagates into model action values $Q(s_t, \cdot)$. In dialogue $Q(s_t, \cdot)$ represent the dialogue action values, which are the expected returns the dialogue system will receive after taking an action under the state s_t . Since $Q(s_t, \cdot)$ are biased, the dialogue policy cannot issue accurate actions accordingly. This hurts dialogue performances.

Example. We use a dialogue turn to show the negative effects of the overestimation bias. In Figure 1b, the dialog state tracker module outputs state embedding, dialogue policy processes state embedding and predicts the wrong dialogue action B instead of the correct action A based on the biased action values.

4 Method

4.1 Dynamic Partial Average

Q-learning suffers from overestimation bias because of the ME (Hasselt, 2010). To reduce the bias, in this work, we propose the dynamic partial average (DPAV) estimator. DPAV utilizes the partial average between the predicted maximal action value and the minimal action value to estimate the ground truth maximal action value $Q_*(s_{t+1})$ of the target of Q-learning update, The mathematical formula of the DPAV estimator of Q-learning is as follows:

$$Q_*(s_{t+1}) \approx (1 - \lambda_t) * \max_{a'} Q(s_{t+1}, a'; \theta') + \lambda_t * \min_{a''} Q(s_{t+1}, a''; \theta') = Q_{DPAV}(s_{t+1}), \quad (4)$$

λ_t is a float number between $[0,1]$ that is dynamic in time and problem-dependent such that the DPAV can take the average between the maximum and minimum of the action values. The weights assigned to the maximum and minimum are not the same, so it is a partial average.

The DPAV estimator is deployed in Q-learning as DPAV Q-learning, so that we have the action value function Q update formula as:

$$\begin{aligned} Q(s_t, a_t) &\leftarrow (1 - \alpha_t)Q(s_t, a_t) + \alpha_t y^{DPAV}, \\ y^{DPAV} &= r_{t+1} + \gamma * Q_{DPAV}(s_{t+1}). \end{aligned} \quad (5)$$

where γ is the discount factor for the future action value and α_t is the step size. λ_t in $Q_{DPAV}(s_{t+1})$ decays according to a predefined rate as training progresses, the decay formula is as follows:

$$\lambda_{t+1} = \lambda_t * d, \quad (6)$$

where d is the decay rate that is set to the fixed value in training. So $1 - \lambda_t$ will give more weight to the maximal action value during the training.

To apply DPAV to the complex dialogue policy learning setting, this paper combines it with the deep Q-learning network (DQN) and proposes DPAV DQN. Its loss function is adapted from Equation 2 to the formula:

$$L_\theta = \mathbb{E}_{\langle s_t, a_t, r_{t+1}, s_{t+1} \rangle \sim m} \left[(y^{DPAV} - Q(s_t, a_t; \theta))^2 \right]. \quad (7)$$

The algorithm of the dynamic partial average deep Q-learning network is summarized in Algorithm 1.²

The intuition behind this approach is that the predicted maximal action value overestimates the ground truth, so DPAV uses the predicted minimal action value to shift the estimate towards the ground truth. Because the predicted action values accuracy is improved in training, DPAV assigns less weight to the predicted minimum to avoid shifting towards the small estimate too much. DPAV reduces the overestimation bias in the target of the training loss, so it is less biased. This improves the dialogue action values accuracy of the DPAV DQN dialogue policy, so this dialogue policy issues more accurate dialogue actions accordingly which improve dialogue performances.

²In the algorithm, if the state s_{t+1} is a terminal state, it means the Markov decision process ends. And in the dialogue, it means the dialogue ends.

Additionally, this method has a lower computational complexity compared to those of ensemble models. Even if the latter could trade time complexity for space complexity by parallel computing, they still have high computational complexity in general as shown in the Table 2. And this method achieves better or comparable performances according to the Figure 3. The upper and lower bounds of the DPAV DQN estimation bias are also reasonable compared with those of other methods. A detailed explanation is found in section 4.3.

Algorithm 1: DPAV DQN

Initialize replay memory \mathcal{D} to capacity N , action-value function Q with random weights, and decay rate d

for $episode = 1, \dots, M$ **do**

 Initialize state s_1

for $j=1, \dots, T$ **do**

 With probability ϵ select a random action a_j , otherwise select

$$a_j = \max_a Q^*(s_j, a; \theta)$$

 Execute action a_j in environment, observe reward r_{j+1} and come into state s_{j+1} . Store transition

$(s_j, a_j, r_{j+1}, s_{j+1})$ in \mathcal{D} , and set

$$s_j = s_{j+1}$$

 Sample random minibatch of

$(s_t, a_t, r_{t+1}, s_{t+1})$ from \mathcal{D}

$$\text{Set } y_t = \begin{cases} r_{t+1} & \text{if terminal state } s_{t+1} \\ y^{DPAV} & \text{non-terminal state } s_{t+1} \end{cases}$$

 Perform a gradient descent step on $L_\theta = (y_t - Q(s_t, a_t; \theta))^2$ to update θ

 Replace target parameter $\theta^- \leftarrow \theta$ after every L iterations. Update average weight $\lambda_{t+1} \leftarrow \lambda_t * d$ after every U iterations.

end

end

4.2 Convergence

In this subsection we show in Theorem 1 that in the limit DPAV Q-learning converges to the optimal

policy. The proof³ of this result using the Lemma 1 (Singh et al., 2000) is in the Appendix A.2.

Theorem 1. In a Markov decision process, the approximate action value function Q as updated by DPAV Q-learning in Equation 5 converges to the optimal action value function q_* with probability one if an infinite number of experience tuples in the form of $(s_t, a_t, r_{t+1}, s_{t+1})$ are given by a learning policy for each state action pair and if the following conditions are satisfied:

1. The Markov decision process is finite (i.e. $|\mathcal{S} \times \mathcal{A} \times \mathcal{R}| < \infty$, \mathcal{S} means the set of states, \mathcal{A} means the set of actions, and \mathcal{R} is the set of rewards.).
2. $\gamma \in [0, 1)$.
3. $\alpha_t(s, a) \in [0, 1]$, $\sum_t \alpha_t(s, a) = \infty$, $\sum_t \alpha_t^2(s, a) < \infty$ w.p.1, and $\forall s, a \neq s_t, a_t : \alpha_t(s, a) = 0$. $\alpha_t(s, a)$ is the step size of a Q-learning update.

4.3 Upper and Lower Bound

As shown in (D’Eramo et al., 2016; Hasselt, 2010), considering a set of $M \geq 2$ independent random variables $X = \{X_1, \dots, X_M\}$, each random variable X_i has a mean $\mu_i = \mathbb{E}[X_i]$ and a variance $\sigma_i = \text{Var}[X_i]$. In many problems, one is interested in the maximum expected value in such a set $\mu_* = \max_i \mathbb{E}\{X_i\}$. Without knowledge of the functional form and parameters of the underlying distribution of each variable X_i , it is impossible to find μ_* analytically. Given a set of a limited number of samples, $S = \{S_1, \dots, S_M\}$, S_i corresponds to the subset of samples drawn from the unknown distribution of the random variable X_i . The maximum estimator (Watkins and Dayan, 1992) and double estimator (Hasselt, 2010) are the most representative methods to estimate μ_* . ME estimation: $\hat{\mu}_*^{ME}(S) = \max_i \hat{\mu}_i(S) = \max_i E\{S_i\} \approx \mu_*$. DE splits the set S into $S^A = \{S_1^A, \dots, S_M^A\}$ and $S^B = \{S_1^B, \dots, S_M^B\}$. DE estimation: $\hat{\mu}_*^{DE}(S^A, S^B) = \hat{\mu}_{a^*}(S^B) = E[S_{a^*}^B] \approx \mu_*$, with $a^* = \arg \max_i \hat{\mu}_i(S^A)$.

4.3.1 Bias

We start with representing the main results about the bias of Maximum Estimator (ME) and Double Estimator (DE) reported in (Van Hasselt, 2013). As for the direction of the bias, ME is positively biased, while DE is negatively biased. ME is

bounded by: $\text{Bias}(\hat{\mu}_*^{ME}) \leq \sqrt{\frac{M-1}{M} \sum_{i=1}^M \frac{\sigma_i^2}{|S_i|}}$. For the bound of DE, (Van Hasselt, 2013) conjectures the following lower bound: $\text{Bias}(\hat{\mu}_*^{DE}) \geq -\frac{1}{2} \left(\sqrt{\sum_{i=1}^M \frac{\sigma_i^2}{|S_i^A|}} + \sqrt{\sum_{i=1}^M \frac{\sigma_i^2}{|S_i^B|}} \right)$. M means the number of sample means, σ_i means the variance of the i_{th} sample mean. For the bias of DPAV estimator, we have the following bounds.

Theorem 2. For any given set X of M random variables: $\text{Bias}(\hat{\mu}_*^{DPAV}) \leq \text{Bias}(\hat{\mu}_*^{ME})$, and $\text{Bias}(\hat{\mu}_*^{DPAV}) \geq \text{Bias}(\hat{\mu}_*^{DE})$.

Explanation. ME uses the maximum of sample means to estimate the ground truth maximal expected value (MEV), while DPAV takes the partial average over the maximum and minimum of sample means. The minimum will shift the DPAV estimation towards the ground truth, so the upper bound of the DPAV estimator bias will be smaller than that of ME. DE uses the minimum of sample means to estimate the ground truth in the worst case, however, the DPAV estimator mitigates this bias through importing the maximum into the partial average shifting the estimation towards the ground truth. So its lower bound is larger than that of DE.

4.3.2 Variance

Since the MSE loss of an estimator is the sum of its squared bias and its variance, so we should also consider its variance to evaluate its goodness. Van Hasselt (2013) proved that both the variance of ME and the one of DE could be upper bounded by the sum of variances of sample means: $\text{Var}(\hat{\mu}_*^{ME/DE}) \leq \sum_{i=1}^M \frac{\sigma_i^2}{|S_i|}$.

Theorem 3. The variance of DPAV estimator is upper bounded by: $\text{Var}(\hat{\mu}_*^{DPAV}) \leq \frac{\sigma_{Max/Min}^2}{|S_{Max/Min}|} \leq \sum_{i=1}^M \frac{\sigma_i^2}{|S_i|}$.

Explanation. Because the DPAV estimator utilizes the partial average between the maximum and minimum of sample means to estimate the ground truth. The weights assigned to the maximum and minimum are in the range (0,1), and the sum of weights is 1. According to the variance math properties (Casella and Berger, 2021), the estimation variance is smaller than the larger one among the variances of maximum and minimum of sample means. Therefore, it is also smaller than the maximal variance of all sample means.

³Lemma 1 was also used to prove the convergence of SARSA (Rummery and Niranjan, 1994) and Double Q-learning (Van Hasselt et al., 2016)

Task	Intents	Slots	User goals
Movie-ticket booking	11	16	128
Restaurant reservation	11	30	3525
Taxi ordering	11	29	2830

Table 1: The statistics of the datasets.

5 Experiments

5.1 Dataset and Evaluation Metrics

We evaluate the DPAV DQN method and baselines on three public task-completion dialogue datasets⁴: movie-ticket booking (Li et al., 2016, 2017), restaurant reservation and taxi ordering (Li et al., 2018). The statistics of the datasets are given in Table 1 (see Appendix B.1 for details)

The evaluation metrics are **success rate** and **averaged reward**. Success rate is the ratio of the number of tasks successfully completed by the dialogue system in evaluation to the total number of dialogues in the test set. Averaged reward refers to the average of the cumulative rewards obtained by the dialogue system for completing each dialogue of the test set.

5.2 Baselines

To benchmark our method performance, we use different DQN variants as baselines in dialogue policy module for comparison: (1) **DQN** policy is learned with standard DQN algorithm (Mnih et al., 2015). (2) **Duel DQN** policy is learned by the duel network structure (Wang et al., 2016). (3) **Double DQN** policy uses Double Estimator of Q-learning to train (D’Eramo et al., 2016). (4) **Averaged DQN** policy is trained by taking average over multiple action values of target networks (Anschel et al., 2017). (5) **Maxmin DQN** policy uses the minimum of multiple maximums from different ensemble units to estimate the ground truth maximal action value in a selective process (Lan et al., 2020). (6) **SUNRISE** policy trains with weighted Bellman backups from multiple networks (Lee et al., 2021). Our model **DPAV** uses a value function instead of a combination of multiple value functions to tailor the maximum and the minimum action value.

We conduct two λ searching schemes: neural network (NN) searching and heuristic searching. We also analyze the influence of different initial value λ_0 in the heuristic searching. So we have following models in the experiment: (1) **LambdaX** is

⁴(Zhao et al., 2021) argued that the three tasks have been widely used in the research of dialogue policy.

the heuristic searching version of the DPAV DQN. The floating number X is the initial value λ_0 with the range $(0, 1)$. And **LambdaX** (e.g. Lambda0.5, Lambda0.6) searches different floating numbers X for initial λ_0 in the heuristic searching. (2) **LambdaNet** is the neural network searching version of the DPAV DQN. It trains a NN to find a value for λ_t for each dialogue state s_k in the training process. Here, λ_t means the value λ in the training episode t , and s_k represents the dialogue state k sampled from the experience replay buffer of reinforcement learning.

5.3 Implementation Details

This work is implemented with PyTorch toolkit. Compared with the standard DQN algorithm, we change the loss with the one defined by DPAV DQN in Algorithm 1. For these RL-based dialogue policies, action value network $Q(\cdot)$ is a MLP with one hidden layer of 80 hidden nodes. ReLU is the activation function. A greedy policy is used in the evaluation. All neural networks warm start 120 episodes using the same rule-based policy before training and are trained with the same hyper-parameters. We follow the default hyper-parameters of the user simulator setting. The discount factor γ for future reward is 0.9. The batch size is 16, and the learning rate is 0.001. The test set size in the movie domain and other domains is set to 100 and 500, respectively. All baselines are based on DQN for a fair comparison. We set $L = 40$ as the maximum of dialogue turns in all domains. The heuristically searched decay rate d and decay interval of the DPAV estimator in the movie domain and other domains are set to $(0.75, 15$ train iterations) and $(0.9965, 30$ train iterations), respectively. For specific parameters of each model and the user simulator, we refer to Appendix B.2.

5.4 Main Results

The main simulation results are reported in Figure 3, we evaluate each dialogue policy performance in terms of success rate and averaged reward. The top two rows of Figure 3 show DPAV DQN consistently outperforms DQN. The overestimation error in target Q values gets propagated into the DQN Q values, while DPAV DQN reduces the overestimation error then its Q values will be less biased. So it more correctly creates dialogue utterances based on the Q values and achieves a better success rate and averaged reward.

Our DPAV DQN method performs better than

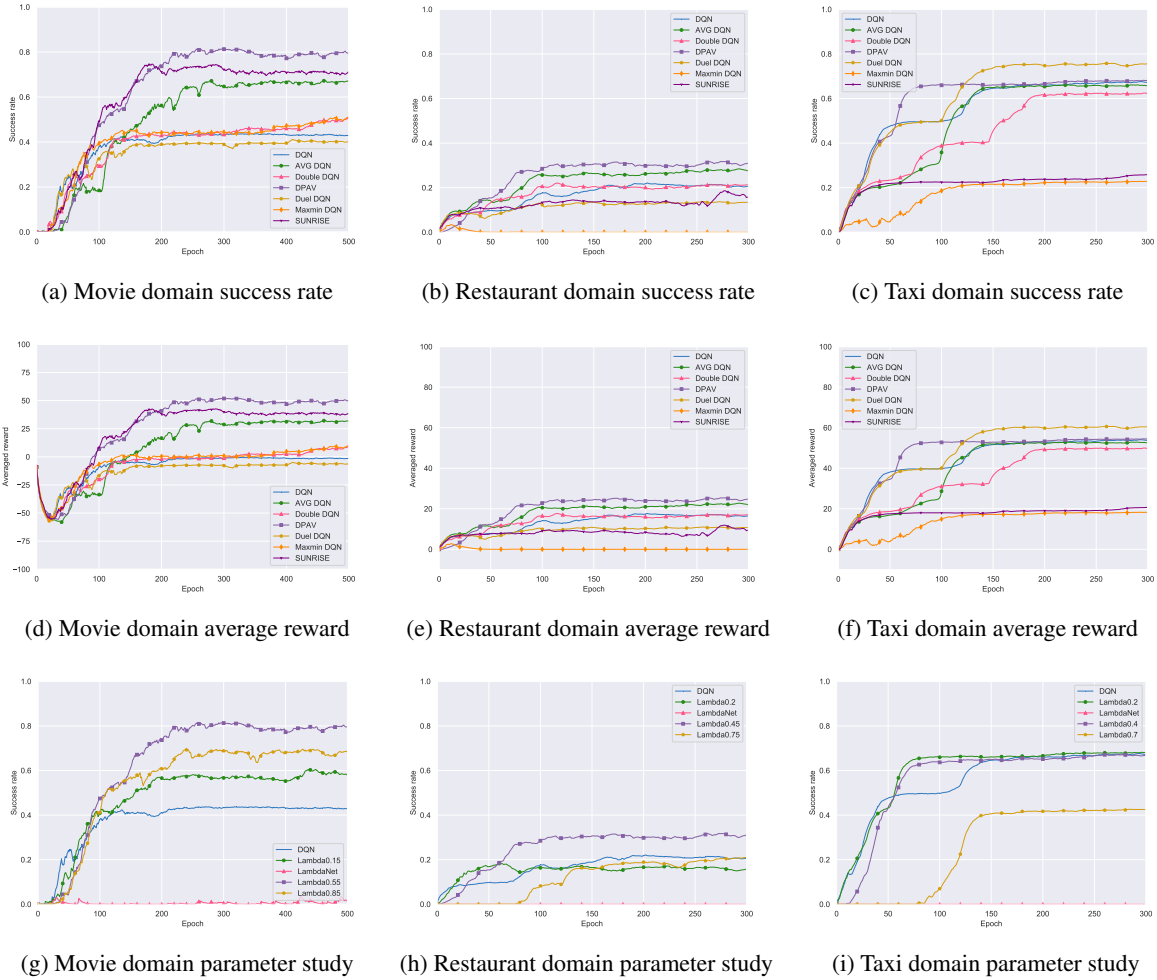


Figure 3: The **top** row shows the learning curves of dialogue policies. The X-axis is the number of training epochs and the Y-axis is the success rate of dialogue policies on the test dataset. The **second** row shows the averaged reward of each dialogue in the test dataset. The **third** row shows the influences of different initial λ values and value search schemes. The X-axis and Y-axis are the same as those of the top row. Each learning curve is averaged over 3 runs on the test dataset.

the baselines in terms of general performance. Since the training starts with the experience pool initialized by the the same rule-based dialogue policy, the models' performance in the very first few episodes is very similar. After that, the performance improved for all models, but much rapidly for DPAV DQN, which finally converges to a higher success rate and averaged reward. As we claimed above, the DPAV estimator reduces the overestimation error propagated into its model Q values and results in better action values estimation. Ensemble models performance relies on its number of networks. With a limited number of networks, as mentioned in the Related Work, Averaged DQN still suffers overestimation bias, Maxmin DQN has estimation bias from the coarse estimator and SUNRISE only down-weights the biased estimation. For

non-ensemble models, Duel DQN suffers overestimation with the Maximum Estimator, while Double DQN has underestimation bias (Anschel et al., 2017). These drawbacks of the baselines get their biased loss propagated into the policy model Q values and hurt the accuracy of the policy models. So their performance (i.e., success rate and averaged reward) cannot improve further after reaching a certain level. The training efficiency and performance of DPAV DQN in comparison validate the effectiveness of our model.

However, in the taxi domain, Duel DQN outperforms other dialogue policies. DPAV DQN only slightly improves the results compared to DQN but it converges faster than DQN. Because sometimes there is no explicitly preferable action for a state, so the action values of the state will be sim-

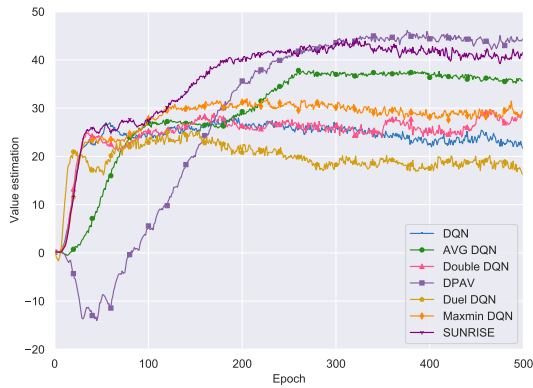


Figure 4: The learning curves of the averaged maximal action value of the dialogue starting state when dialogue policies are evaluated on the movie test set during the training. The Y-axis means the averaged maximal action value of the starting state.

ilar (Thrun and Schwartz, 1993), and the DPAV estimator cannot notably reduce estimation bias through averaging between maximum and minimum action values. But the DPAV estimator still could estimate better than other baselines (except for Duel DQN) as shown in the results. Duel DQN uses the duel network structure to estimate action values (Wang et al., 2016), which is helpful for recognizing the correct action when confronted with confusing states.

5.5 Influence of Parameter λ

Intuitively, the optimal λ should seek the best trade-off between the estimated maximum and minimum that could be used to train the dialogue policy properly. It is a non-trivial optimization problem because the distribution of action values $Q(s_t, \cdot)$ at state s_t is constantly updated, and the optimal λ for s_t should be adjusted accordingly. The third row of Figure 3 shows that with the neural network (NN) searching almost each dialogue policy evaluation has zero success rate and can not converge. Since the distribution and the optimal λ are changing for the same state s_t , the fixed λ searched by the neural network does not work. This validates that the λ for s_t is dynamic, and a fixed λ leads to bad performances.

It is a difficult problem to calculate the exact ground truth maximal action value, so existing works use estimators to approximate it (Lan et al., 2020; Lee et al., 2021; Anschel et al., 2017). DPAV DQN uses the DPAV estimator for the approximation. In the heuristic searching version of DPAV

DQN, λ_0 is a very important initial value for the DPAV estimator. λ_0 is the initial weight of the minimum, because finally we will give the total confidence to our model, the weight of the maximum will be nearly 1. So the lower bound for λ_0 is 0. Since model reduces the overestimation of the maximum through shifting the estimate towards the minimum action value. It is a trade-off, so the upper bound for λ_0 is 1. It is problem dependent and should be set in a range (0,1).

As shown in the third row of Figure 3, among three dialogue datasets: movie, restaurant and taxi, we empirically find that the initial value λ_0 around 0.5 results in good performances. And other heuristic values degrade the dialogue policy performances. Since shifting the estimate towards the minimum action value too much or too less both causes the estimation bias of the ground truth maximal action value. These validate that λ_0 is problem dependent and λ_t should decay to proper values to balance the maximum and the minimum along the training.

5.6 Computational Complexity Comparison

All baselines and DPAV DQN use various estimators or estimation tricks to approximate the ground truth maximal action value $Q_*(s_{t+1})$. Given the state s_{t+1} as the input to all baselines and DPAV DQN, the time complexity of the forward propagation of every model unit has a similar time complexity besides the minor differences (e.g., addition). In order to facilitate the comparison, we denote the time complexity for the forward propagation of every model unit as $O(N)$, here, N means the dimension of the vector input for forward propagation. In this comparison, we suppose each ensemble model has K model units.

Combining the results of Figure 3 and Table 2, DPAV DQN achieves better or comparable performances with a lower time complexity. Although the time complexity of ensemble models can be reduced by parallel computing, but that increases the space complexity. So, the overall computational complexity is still high and resource consuming.

5.7 Results on Maximum Action Value

In reinforcement learning, the action value Q is the expectation of return R_t that is the sum of the discounted rewards: $Q(s, a) = E\{R_t | s_t = s, a_t = a\} = E\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\}$. Fig-

Model Name	Time Complexity
DQN	$O(N)$
Double DQN	$O(N)$
Duel DQN	$O(N)$
Averaged DQN	$O(K*N)$
Maxmin DQN	$O(K*N)$
SUNRISE	$O(K*N)$
DPAV DQN	$O(N)$

Table 2: Time complexity comparison among baselines and DPAV DQN. N refers to the dimension of the vector input for the forward propagation. K means the number of ensemble units. O measures the time complexity of models.

Figure 4 shows learning curves⁵ of the averaged maximum action value for the starting state on the test set, the value in dialogue context means how much return the dialogue policy assumes it could maximally receive from the starting state.

At the first few training epochs, we notice that the averaged maximum value of DPAV DQN is negative which is consistent with the averaged reward of its evaluation shown in Figure 3d, because at the early training stage, the policy quality is too low to finish most of the dialogues so the averaged reward is low and the averaged maximum action value should be low if the model Q values are accurate. But the values of other models are not consistent with and larger than the real averaged reward. Because the estimation bias of the loss makes that these models have inaccurate Q values, the maximum action value of these models is larger than the ground truth.

The policy training based on these inaccurate Q values will be negatively affected. Only using Maximum Estimator (ME) will cause overestimation bias and even lead to worse policy quality, it can be observed from the curves of DQN and Duel DQN in Figure 4. Averaged DQN and Maxmin DQN use ME in their single unit so the bias leads their Q functions to converge into inaccurate values, which prevents averaged maximum action values from improving further. SUNRISE down-weights the biased estimation and it is trained in such a way so that SUNRISE dialogue policy receives more rewards during the evaluation 3d. As shown in the Figure 4, the averaged maximal action value of DPAV DQN remains the highest among the three datasets because its model gets trained better with

⁵To save space, we only present the results on the movie dataset, and the results on other datasets are similar.

the less biased loss and receives more return from successful dialogues during evaluation. This also coincides with the averaged reward from test dialogues in Figure 3d. This empirically validates that DPAV is a better estimator than others because of less estimation bias.

6 Conclusion

This paper is the first to investigate the negative effects of the overestimation problem in task-completion dialogue systems. We propose the DPAV estimator to mitigate this problem of Q-learning. We also theoretically prove convergence and derive the upper and lower bounds of the estimation bias compared with those of other methods. The resulting DPAV DQN model is empirically evaluated on three dialogue datasets and achieves better or comparable results with lower computational load compared to state-of-the-art baselines.

Acknowledgements

We would like to thank Xiumei Zhao, Dan Li, Wentao Huang and Pengjie Ren for reading the paper draft. This research was partially supported by the China Scholarship Council. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors. Marie-Francine Moens is supported by the ERC Advanced Grant CALCULUS (788506).

References

- Oron Anshel, Nir Baram, and Nahum Shimkin. 2017. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *International conference on machine learning*, pages 176–185. PMLR.
- Shamama Anwar and K Sridhar Patnaik. 2008. Actor critic learning: A near set approach. In *International Conference on Rough Sets and Current Trends in Computing*, pages 252–261. Springer.
- George Casella and Roger L Berger. 2021. *Statistical inference*. Cengage Learning.
- Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35.
- Carlo D’Eramo, Alessandro Nuarra, Matteo Pirota, and Marcello Restelli. 2017. Estimating the maximum expected value in continuous reinforcement learning problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

- Carlo D’Eramo, Marcello Restelli, and Alessandro Nuara. 2016. Estimating maximum expected value through gaussian approximation. In *International Conference on Machine Learning*, pages 1032–1040. PMLR.
- Jianqing Fan, Zhaoran Wang, Yuchen Xie, and Zhuoran Yang. 2020. A theoretical analysis of deep q-learning. In *Learning for Dynamics and Control*, pages 486–489. PMLR.
- Hado Hasselt. 2010. Double q-learning. *Advances in neural information processing systems*, 23:2613–2621.
- Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. 2020. Controlling over-estimation bias with truncated mixture of continuous distributional quantile critics. In *International Conference on Machine Learning*, pages 5556–5566. PMLR.
- Qingfeng Lan, Yangchen Pan, Alona Fyshe, and Martha White. 2020. Maxmin q-learning: Controlling the estimation bias of q-learning. *arXiv preprint arXiv:2002.06487*.
- Donghun Lee, Boris Defourny, and Warren B Powell. 2013. Bias-corrected q-learning to control max-operator bias in q-learning. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 93–99. IEEE.
- Donghun Lee and Warren B Powell. 2012. An intelligent battery controller using bias-corrected q-learning. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.
- Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. 2021. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning*, pages 6131–6141. PMLR.
- Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-end task-completion neural dialogue systems. *arXiv preprint arXiv:1703.01008*.
- Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*.
- Xiujun Li, Yu Wang, Siqi Sun, Sarah Panda, Jingjing Liu, and Jianfeng Gao. 2018. Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *arXiv preprint arXiv:1807.11125*.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.
- Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, Kam-Fai Wong, and Shang-Yu Su. 2018. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. *arXiv preprint arXiv:1801.06176*.
- Andreas Pentaliotis and Marco A Wiering. 2021. Variation-resistant q-learning: Controlling and utilizing estimation bias in reinforcement learning for better performance. In *ICAART (2)*, pages 17–28.
- Gavin A Rummery and Mahesan Niranjana. 1994. *Online Q-learning using connectionist systems*, volume 37. Citeseer.
- Satinder Singh, Tommi Jaakkola, Michael L Littman, and Csaba Szepesvári. 2000. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine learning*, 38(3):287–308.
- James E Smith and Robert L Winkler. 2006. The optimizer’s curse: Skepticism and postdecision surprise in decision analysis. *Management Science*, 52(3):311–322.
- Sebastian Thrun and Anton Schwartz. 1993. Issues in using function approximation for reinforcement learning. In *Proceedings of the Fourth Connectionist Models Summer School*, pages 255–263. Hillsdale, NJ.
- Hado Van Hasselt. 2013. Estimating the maximum expected value: an analysis of (nested) cross validation and the maximum sample average. *arXiv preprint arXiv:1302.7175*.
- 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.
- Sihan Wang, Kaijie Zhou, Kunfeng Lai, and Jianping Shen. 2020. Task-completion dialogue policy learning via monte carlo tree search with dueling network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3461–3471.
- Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. 2016. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR.
- Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning*, 8(3-4):279–292.
- Yuexin Wu, Xiujun Li, Jingjing Liu, Jianfeng Gao, and Yiming Yang. 2019. Switch-based active deep dyna-q: Efficient adaptive planning for task-completion dialogue policy learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7289–7296.
- Rui Zhang, Zhenyu Wang, Mengdan Zheng, Yangyang Zhao, and Zhenhua Huang. 2021. Emotion-sensitive deep dyna-q learning for task-completion dialogue policy learning. *Neurocomputing*, 459:122–130.

Yichi Zhang, Zhijian Ou, Min Hu, and Junlan Feng. 2020a. A probabilistic end-to-end task-oriented dialog model with latent belief states towards semi-supervised learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9207–9219.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020b. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, pages 1–17.

Zhirui Zhang, Xiujun Li, Jianfeng Gao, and Enhong Chen. 2019. Budgeted policy learning for task-oriented dialogue systems. *arXiv preprint arXiv:1906.00499*.

Zongzhang Zhang, Zhiyuan Pan, and Mykel J Kochenderfer. 2017. Weighted double q-learning. In *IJCAI*, pages 3455–3461.

Yangyang Zhao, Zhenyu Wang, Changxi Zhu, and Shihan Wang. 2021. Efficient dialogue complementary policy learning via deep q-network policy and episodic memory policy. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4311–4323.

A Appendix

A.1 Lemma

Lemma 1 (Hasselt, 2010). Let (β_t, Δ_t, F_t) be a stochastic process, where $\beta_t, \Delta_t, F_t : X \mapsto \mathbb{R}$ satisfy,

$$\Delta_{t+1}(x_t) = (1 - \beta_t(x_t)) \Delta_t(x_t) + \beta_t(x_t) F_t(x_t)$$

where $x_t \in X$ and $t = 0, 1, 2, \dots$. Let P_t be a sequence of increasing σ -fields such that β_0 and Δ_0 are P_0 -measurable and β_t, Δ_t , and F_{t-1} are P_t -measurable, with $t \geq 1$. Assume that the following conditions are satisfied:

1. The set X is finite (i.e. $|X| < \infty$).
2. $\beta_t(x_t) \in [0, 1]$, $\sum_t \beta_t(x_t) = \infty$, $\sum_t \beta_t^2(x_t) < \infty$ w.p. 1, and $\forall x \neq x_t : \beta_t(x) = 0$. $\beta_t(x_t)$ is the step size of the update.
3. $\|\mathbb{E}\{F_t | P_t\}\| \leq \kappa \|\Delta_t\| + c_t$, where $\kappa \in [0, 1)$ and $c_t \rightarrow 0$ w.p.1.
4. $\mathbb{V}\{F_t(x_t) | P_t\} \leq C(1 + \kappa \|\Delta_t\|)^2$, where C is some constant.

where $\mathbb{V}\{\cdot\}$ denotes the variance and $\|\cdot\|$ denotes the maximum norm. Then Δ_t converges to zero with probability one.

Proof. See (Singh et al., 2000).

A.2 DPAV Q-learning Convergence Proof

Proof. We apply Lemma 1 with $X = \mathcal{S} \times \mathcal{A}$, $\Delta_t = Q_t(s_t, a_t) - q_*(s_t, a_t)$, $\beta_t = \alpha_t$, β_t is also the step size, $P_t = \{Q_0, s_0, a_0, \alpha_0, r_1, s_1, \dots, s_t, a_t\}$ and

$$F_t(s_t, a_t) = r_{t+1} + \gamma Q_{dpav}(s_{t+1}, \cdot) - q_*(s_t, a_t), \quad (8)$$

where

$$Q_{dpav}(s_{t+1}, \cdot) = (1 - \lambda_t) Q_t(s_{t+1}, a_{\max}) + \lambda_t Q_t(s_{t+1}, a_{\min}). \quad (9)$$

And $a_{\max} = \arg \max_{a'} Q_t(s_{t+1}, a')$ while $a_{\min} = \arg \min_{a''} Q_t(s_{t+1}, a'')$. The first condition of the Lemma 1 is satisfied because $|\mathcal{S} \times \mathcal{A}| < \infty$. The second condition of Lemma 1 is met by the third condition of Theorem 1. Because the absolute value of reward $|r| < \infty \implies \forall t : \mathbb{V}\{r_{t+1} | P_t\} < \infty$. Since Q_t is the expected cumulative reward in Q-learning and $F_t(s_t, a_t)$ is composed of reward r , so $\forall t : \mathbb{V}\{r_{t+1} | P_t\} < \infty \implies \forall t : \mathbb{V}\{F_t(s_t, a_t) | P_t\} < \infty$, the fourth condition of the Lemma 1 is sufficed. This leaves to show that the third condition of the Lemma 1 on the expected contraction of F_t holds. We can write

$$\begin{aligned} F_t(s_t, a_t) &= r_{t+1} + \gamma((1 - \lambda_t) Q_t(s_{t+1}, a_{\max}) \\ &\quad + \lambda_t Q_t(s_{t+1}, a_{\min})) - q_*(s_t, a_t) \\ &= r_{t+1} + \gamma(Q_t(s_{t+1}, a_{\max}) - \lambda_t Q_t(s_{t+1}, a_{\max}) \\ &\quad + \lambda_t Q_t(s_{t+1}, a_{\min})) - q_*(s_t, a_t) \\ &= r_{t+1} + \gamma Q_t(s_{t+1}, a_{\max}) - q_*(s_t, a_t) \\ &\quad + \gamma \lambda_t (Q_t(s_{t+1}, a_{\min}) - Q_t(s_{t+1}, a_{\max})) \\ &= r_{t+1} + \gamma Q_t(s_{t+1}, a_{\max}) - q_*(s_t, a_t) \\ &\quad + \gamma \lambda_t Q_{sub} \\ &= F'_t(s_t, a_t) + \gamma \lambda_t Q_{sub}, \end{aligned}$$

where F'_t is the value of F_t if normal Q-learning would be under consideration, and $Q_{sub} = Q_t(s_{t+1}, a_{\min}) - Q_t(s_{t+1}, a_{\max})$. Since it is well known that $\forall t : \|\mathbb{E}\{F'_t | P_t\}\| \leq \gamma \|\Delta_t\|$ (Hasselt, 2010), it follows that, $\|\mathbb{E}\{F_t | P_t\}\|$

$$\begin{aligned} &= \|\mathbb{E}\{F'_t | P_t\}\| + \gamma \lambda_t \|\mathbb{E}\{Q_{sub} | P_t\}\| \\ &\leq \gamma \|\Delta_t\| + \gamma \lambda_t \|\mathbb{E}\{Q_{sub} | P_t\}\| \end{aligned}$$

Since in DPAV Q-learning, the λ_t will decay as $\lambda_{t+1} = \lambda_t * d$. When $t \rightarrow \infty$, given $\varepsilon > 0, \exists t_0 : \forall t \geq t_0 \implies \lambda_t < \varepsilon \implies \lim_{t \rightarrow \infty} \lambda_t = 0$. Therefore, it suffices to show that $c_t = \gamma \lambda_t Q_{sub} \rightarrow 0$ w.p.1. Since all the conditions of lemma 1 are satisfied, it holds that, $\forall s, a : Q_t(s, a) \rightarrow q_*(s, a)$ w.p.1.

B Appendix

B.1 Dataset details

Table 1 lists the number of intents, slots and users goals in the three datasets used in the evaluation. And Table 3 shows all annotated dialogue acts and slots in details. Task-oriented dialogue systems are designed to help users to complete a specific goal G . Even though the dialogue system knows nothing about the user goal explicitly, the whole dialogue progresses around this user goal G implicitly. In order to explain the user goal better, we take a user goal as an example from the movie domain:

$$\text{Goal} = \left(\begin{array}{l} C = \left[\begin{array}{l} \text{moviename} = \text{Enter} \\ \text{the Dragon} \\ \text{actor} = \text{BruceLee} \\ \text{date} = \text{today} \end{array} \right], \\ R = \left[\begin{array}{l} \text{theater} = \\ \text{starttime} = \end{array} \right] \end{array} \right). \quad (10)$$

In this user goal, a user inquires the dialogue system about the theater and start time of a today’s movie about the Enter the Dragon by Bruce Lee. The user goals are generated from the annotated datasets mentioned in Table 3. The user goals extracted from the same dataset are then aggregated into a user goal set for that task. The user goals extracted from the same dataset are then aggregated into a user goal set for that task. When running a dialogue, the user simulator (Li et al., 2016) randomly samples a user goal from the user goal set to converse with the dialogue system. Helping the user to achieve specific user goals is the task to complete for dialogue systems. In this paper, we use the success rate and averaged reward as our main evaluation criteria. We do not use averaged turns into our criteria because overestimation bias mainly prevents the dialogue system from completing a task in a dialogue. This is explicitly with success rate and averaged reward, and this is not directly related with averaged turns. If and only if the dialogue system recognizes all constraints provided by users and informs all information that users want, and finally books the desired tickets successfully, the user goal is viewed as successful, and the dialogue policy received positive reward for success. The averaged reward means the averaged cumulative discounted reward received by dialogue system per dialogue.

B.2 Implementation details

The size of the experience replay pool in the movie domain and other domains is set to 8000 and 10000, respectively. The number of target networks in Averaged DQN, Maxmin DQN and SUNRISE is set to 4. The temperature parameter of SUNRISE is set to 2. The target network update period for Averaged DQN is set to 4. In the experiment, we use a user simulator to interact with dialogue systems. In the movie domain, the dialogue system receives a 2L reward if the dialogue finishes successfully and receives -L if it fails. Also, a fixed reward (-1) is given to the dialogue system for each dialogue turn. In the restaurant and taxi domains, the dialogue system receives a 2L reward if the dialogue finishes successfully and receives 0 if it fails. Also, a fixed reward (0) is given to the dialogue system for each dialogue turn. Under this setup, the dialogue datasets for experiments have varieties (Li et al., 2016).

Task	Intents	Slots	Dialogues
Movie	request,inform, confirm_question, confirm_answer, greeting,closing, deny,not_sure, multiple_choice, thanks,welcome	city,closing, data,greeting, distanceconstraints, moviename,price, numberofpeople, starttime,state, taskcomplete,theater, teater_chain,ticket, video_format,zip	280
Restaurant	request,inform, confirm_question, confirm_answer, greeting,closing, deny,not_sure, multiple_choice, thanks,welcome,	address,atmosphere, choice,city,closing, cuisine,date,food, dress_code,greeting, distanceconstraints, numberofkids,mealtype, numberofpeople, other,personfullname, phonenumber,pricing, rating,restaurantname, restauranttype,seating, starttime,state,zip, result,occasion, taskcomplete,reservation	4103
Taxi	request_inform, comfirm_question, confirm_answer, greeting,closing, deny,not_sure, multiple_choice, thanks,welcome,	car_type,city,speed, closing,car_level,date, distanceconstraints, dropoff_location, zip,result,numberofkids, greeting,name,driver_id, numberofpeople,other, pickup_location,state, dropoff_location_city, pickup_location_city, pickup_time,cost, taxi_company,mc_list, taskcomplete,taxi,budget, emergency_degree,drive_level	3094

Table 3: The details of the datasets. (Li et al., 2016, 2018)