Learning Efficient Dialogue Policy from Demonstrations through Shaping

Huimin Wang[♣†] **Baolin Peng**[♠†*] **Kam-Fai Wong**[♣]
[♠]Microsoft Research
[♣]The Chinese University of Hong Kong
bapeng@microsoft.com
{hmwang,kfwong}@se.cuhk.edu.hk

Abstract

Training a task-oriented dialogue agent with reinforcement learning is prohibitively expensive since it requires a large volume of interactions with users. Human demonstrations can be used to accelerate learning progress. However, how to effectively leverage demonstrations to learn dialogue policy remains less explored. In this paper, we present that efficiently learns dialogue policy from demonstrations through policy shaping and reward shaping. We use an imitation model to distill knowledge from demonstrations, based on which policy shaping estimates feedback on how the agent should act in policy space. Reward shaping is then incorporated to bonus state-actions similar to demonstrations explicitly in value space encouraging better exploration. The effectiveness of the proposed S^2*Agent* is demonstrated in three dialogue domains and a challenging domain adaptation task with both user simulator evaluation and human evaluation.

1 Introduction

With the flourishment of conversational assistants in daily life (like Google Assistant, Amazon Alexa, Apple Siri, and Microsoft Cortana), task-oriented dialogues that are able to serve users on certain tasks have increasingly attracted research efforts. Dialogue policy optimization is one of the most critical tasks of dialogue modeling. One of the most straightforward approaches is the rule-based method, which contains a set of expert-defined rules for dialogue modeling. Though rule-based dialogue systems have a reasonable performance in some scenarios, handcrafting such kinds of rules is time-consuming and not scalable.

Recently, dialogue policy learning is formulated as a reinforcement learning (RL) problem and tackled with deep RL models (Li et al., 2017; Lipton et al., 2018; Peng et al., 2017). It has shown great potentials of using the RL-based method for building robust dialogue systems automatically. However, due to its interactive nature, RL-based agents demand of an environment to operate in. As illustrated in Figure 1, RL-based dialogue agents need to interact with human users and update its policy in an online fashion requiring that the agents have a good online performance from the start of training. In addition, one of the biggest challenges of RL approaches is reward sparsity issue, which leads to exploration in large action space inefficient. As a consequence, training RL-based agents expects a prohibitively large number of interactions to achieve acceptable performance, which may incur a significant amount of expense (Pietquin et al., 2011; Lipton et al., 2016; Peng et al., 2018b). Several attempts are made to improve learning efficiency and tackle reward sparsity issues. Different types of heuristics has been proposed in the form of intrinsic rewards to guide exploration more efficiently (Lipton et al., 2016; Mohamed and Rezende, 2015; Peng et al., 2017, 2018a; Takanobu et al., 2019).

When building a dialogue system, it is typically affordable to recruit experts to gather some demonstrations about the expected agent behaviors. We therefore aim to address the aforementioned challenges from a different perspective and assume having access to human-provided demonstrations. In this paper, we investigate how to efficiently leverage these demonstrations to alleviate reward sparsity and improve policy learning quality. Previous work (Lipton et al., 2016) used a simple technique termed as Replay Buffer Spiking (RBS) to pre-fill experience replay buffer with human demonstrations, which yields good performance, especially in the beginning. (Hester et al., 2018) proposed Deep Q-learning from Demonstrations (DQfD) that combines temporal difference updates with a supervised classification loss of actions in demonstrations to

*Corresponding author †Equal Contribution

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6355–6365

improve learning efficiency in gaming domains. However, whether it is feasible and how to effectively leverage human demonstration in dialogue scenarios are less explored.

Hence, in this paper, we propose a new strategy of leveraging human demonstrations to learn dialogue policy efficiently. Our dialogue agent, termed as S^2*Agent*[1], learns dialogue policy from demonstrations trough *policy shaping* and *reward shaping*. Policy shaping (Griffith et al., 2013) is an approach to incorporating human feedback to advise how policy should behave like experts. It estimates feedback of a state-action pair from human demonstrations and then utilizes the feedback to reconcile the policy from any RL-based agents. This method speeds up learning progress in gaming domains but has not yet been studied in dialogue. However, directly applying policy shaping to dialogue faces several challenges. The original policy shaping uses a tabular analogous method to estimate feedback. This method limits its feasibility for complex problems like dialogue that has large state action representations. To deal with this issue, we propose to use deep neural networks, which represent state-action space with function approximation and distill knowledge from human demonstrations, to estimate feedback. In addition, policy shaping calibrates agents' behavior in policy space, and it is inherently not designed to tackle reward sparsity issues. Considering this, we further introduce *reward shaping* to bonus these state-action pairs that are similar to demonstrations. It can be viewed as a shaping mechanism explicitly in value space to guide policy exploration towards actions which human experts likely conduct. Our contributions in this work are two-fold:

- We propose a novel S^2*Agent* that can effectively leverage human demonstrations to improve learning efficiency and quality through policy shaping and reward shaping.
- We experimentally show that S^2*Agent* can efficiently learn good policy with limited demonstrations on three single domain dialogue tasks and a challenging domain adaptation task using both simulator and human evaluations.

[1]Agent with policy **S**haping and reward **S**haping

2 Related Work

Dialogue policy learning Deep reinforcement learning (RL) methods have shown great potential in building a robust dialog system automatically (Young et al., 2013; Su et al., 2016; Williams et al., 2017; Peng et al., 2017, 2018a,b; Lipton et al., 2018; Li et al., 2020; Lee et al., 2019). However, RL-based approaches are rarely used in real-world applications, for these algorithms often require (too) many experiences for learning due to the sparse and uninformative rewards. A lot of progress is being made towards mitigating this sample complexity problem by incorporating prior knowledge. (Su et al., 2017) utilizes a corpus of demonstration to pre-train the RL-based models for accelerating learning from scratch. (Chen et al., 2017b) attempts to accelerate RL-based agents by introducing extra rewards from a virtual rule-based teacher. However, the method requires extra efforts to design a rule-based dialogue manager. (Hester et al., 2018) improve RL learning by utilizing a combination of demonstration, temporal difference (TD), supervised, and regularization losses. (Chen et al., 2017a) introduced a similar approach called companion teaching to incorporate human teacher feedback into policy learning. Nevertheless, companion teaching assumes that there is a human teacher to directly give a correct action during policy learning process and meanwhile train an action prediction model for reward shaping based on human feedback.

Policy shaping Policy Shaping is an algorithm that enables introducing prior knowledge into policy learning. (Griffith et al., 2013) formulates human feedback on the actions from an agent policy as policy feedback and proposes Advise algorithm to estimate humans Bayes feedback policy and combine it with the policy from the agent. It shows significant improvement in two gaming environment. (Misra et al., 2018) uses policy shaping to bias the search procedure towards semantic parses that are more compatible with the text and achieve excellent performance.

Reward shaping Reward shaping leverages prior knowledge to provides a learning agent with an extra intermediate reward F in addition to environmental reward r, making the system learn from a composite signal $R + F$ (Ng et al., 1999). However, it is not guaranteed that with reward shaping, an MDP can still have an optimal policy that is

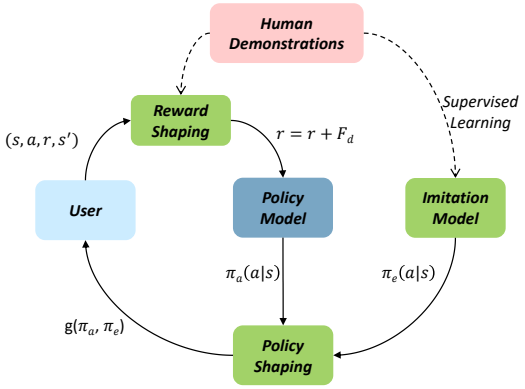


Figure 1: Illustration of the S^2Agent for dialogue policy learning.

identical to the original problem unless the shaping is potential-based reward shaping(Ng et al., 1999; Marthi, 2007). (Su et al., 2015) proposes to use RNNs to predict turn-level rewards and use the predicted reward as informative reward shaping potentials. (Peng et al., 2018a; Takanobu et al., 2019) use inverse reinforcement learning to recover reward functions from demonstrations for reward shaping. However, the estimated reward using these methods inevitably contains noise and failed to conform to potential-based reward function to guarantee the optimal policy. Inspired by (Brys et al., 2015), we directly estimate potential-based reward function from demonstrations.

3 Approach

Our *S^2Agent* is illustrated in Figure 1, consisting of four modules. 1) Dialogue policy model which selects the best next action based on the current dialogue state.; 2) Imitation Model is formulated as a classification task that takes dialogue states as input and predicts associated dialogue action, aiming to distill behaviors from human demonstrations.; 3) Policy Shaping provides feedback on how policy should behave like demonstrations. It then reconciles a final action based on actions from the policy model and imitation model attempting to generate more reliable exploration trajectories; 4) Followed by a reward shaping module that encourages demonstration similar state-actions by providing extra intrinsic reward signals.

3.1 Policy Model

We consider dialogue policy learning as a Markov Decision Process (MDP) problem and improve the policy with Deep Q-network (DQN) (Mnih et al., 2015).[2] In each turn, the agent observes the dialogue state s, and then execute the action a with ϵ-greedy exploration that selects a random action with probability ϵ or adopts a greedy policy $a = \text{argmax}_{a'} Q(s, a'; \theta)$, where $Q(s, a'; \theta)$ approximates the value function, implemented as a multi-layer perceptron (MLP) parameterized by θ. The agent then receives the reward r, perceives the next user response to a^u, and updates the state to s'. The tuple (s, a, r, s') is stored in the experience replay D^a. This loop continues until the dialogue terminates. The parameters of $Q(s, a'; \theta)$ are updated by minimizing the following square loss with stochastic gradient descent:

$$\begin{aligned} \mathcal{L}(\theta) = &\ \mathbb{E}_{(s,a,r,s')\sim D^a}[(y_i - Q(s, a; \theta))^2] \\ y_i = &\ r + \gamma \max_{a'} Q'(s', a'; \theta') \end{aligned} \tag{1}$$

where $\gamma \in [0, 1]$ is a discount factor, and $Q(.)$ is the target value function that is only periodically updated (line 26 in Algorithm 1). By differentiating the loss function with regard to θ, we derive the following gradient:

$$\begin{aligned} \nabla_\theta \mathcal{L}(\theta) = E_{(s,a,r,s')\sim D^a}[(r + \\ \gamma \max_{a'} Q'(s', a'; \theta') - \\ Q(s, a; \theta))\nabla_\theta Q(s, a; \theta)] \end{aligned} \tag{2}$$

As shown in lines 25-26 in Algorithm 1, in each iteration, we update $Q(.)$ using minibatch Deep Q-learning.

3.2 Imitation Model

We assume having access to a corpus of human-human dialogues either from a log file or provided by recruited experts, which in this paper are termed as human demonstrations D^e. D^e usually consists of a set of state-action pairs $[(s_1, a_1), (s_2, a_2), ..., (s_n, a_n)]$. Theoretically, if D^e is large enough to cover all the possible states, then the agent can respond perfectly by looking up the corresponding action from D^e.

However, in practice, D^e is usually limited and can not cover all the states. Hence, we propose to use a supervised learning model (denoted as Imitation Model) to parameterize the relation of the

[2]Our shaping methods are compatible with any policy optimizer. In this paper, we employ DQN due to its simplicity and robustness in training. However, replacing with other methods like Actor-Critic is straightforward.

states and actions expecting it to generalize to unseen state. We formulate the task as a classification problem. It takes dialogue s_i as input and is trained with cross-entropy to minimize loss between action a_i and predicted action a. There are multiple models like RNN, CNN can be used for this purpose, but for simplicity, we choose to use MLP.

3.3 Policy Shaping

Incorporating human feedback into RL can accelerate its learning progress (Griffith et al., 2013; Cederborg et al., 2015). Policy shaping is a representative that estimates human's Bayes optimal feedback policy and then combine the feedback policy with the policy of an underlying RL model. The feedback policy is computed with the following equation:

$$\pi_e(a|s) = \frac{\mathcal{C}^{\Delta_{s,a}}}{\mathcal{C}^{\Delta_{s,a}} + (1-\mathcal{C})^{\Delta_{s,a}}} \quad (3)$$

where $\Delta_{s,a}$ is the difference between the number of positive feedback and negative feedback, i.e. the number of occurrence of (s, a) in human demonstrations. $\mathcal{C}$ here means the probability of consistency feedback from demonstrations [3]. For example, $\mathcal{C} = 0.7$ means with 0.7 probability the feedback from the demonstrations is considered reliable. Otherwise, if $\mathcal{C} = 0.5$, then policy shaping is meaningless since it treats every action equally.

However, $\Delta_{s,a}$ is difficult to estimate from the demonstrations in dialogue scenarios since the state and action are large and sparse. To deal with this issue, we propose to use the aforementioned Imitation Model to estimate feedback from demonstrations. Specifically, we samples N times from imitation model policy $\pi_e(a|s)$ to form a committee $a_1, a_2, ..., a_N$ denoting N votes. Then we count for each action to generate c_a as positive feedback from human demonstrations. We use the expectation of binomial distribution $N * (1 - \mathcal{C})$ as the number of negative feedback. Such that, in dialogue, we use:

$$\Delta_{s,a} = c_a - N * (1 - \mathcal{C}) \quad (4)$$

Finally, the policy is reconciled from the policy model and the imitation model by multiplying them together:

$$\pi(a|s) = \frac{\pi_a(a|s) \times \pi_e(a|s)}{\sum_a \pi_a(a|s) \times \pi_e(a|s)} \quad (5)$$

[3] It is a parameter to control noise in the demonstrations.

Policy shaping operates in the policy space and can be viewed as a mechanism of biasing the agent learning towards the policy distilled from the demonstrations to improve learning efficiency. The reconciled policy in equ. 5 allows the underlying RL model surpass the imitation model π_e.

Algorithm 1 *S^2Agent* learining algorithm

Input: $N, \epsilon, \theta, \mathcal{C}, D^a, D^e, \gamma, \mathrm{Z}$
Output: $Q_\theta(s, a)$.
init experience replay D^a as empty.
init $Q_\theta(s, a)$ and $Q'_{\theta'}(s, a)$ with $\theta = \theta'$.
init demo buffer D^e with human conversation data. Train Expert with D^e and load $\pi^e(a|s)$.
for n=1:N **do**
  user starts a dialogue with user action a^u.
  init dialogue state s.
  while s is not terminal **do**
    with probability ϵ select a random action a.
    otherwise select $a = \mathrm{argmax}_a\, Q(s, a; \theta)$.
    #policy shaping starts
    count the number of occurrence for each action and then compute Δ_a with equ.4.
    obtain shaped action distribution from policy shaper following equ.3.
    reconcile the final action distribution as 5 and sample action a.
    #policy shaping ends
    execute a, obtain next state s', receive reward r.
    calculate $\phi_n(s, a)$ with equ.9.
    if $n > 1$ **then**
      #reward shaping starts
      obtain F_D with equ.7.
      Store transition $(s, a, r + F_D, s')$ in D^a
      #reward shaping ends
    end if
  end while
  Sample mini batch of (s, a, r, s') from D^a
  update Q_θ via minibatch Q-learning according to gradient of equ.1.
  every Z steps reset $Q_\theta = Q'_\theta$.
end for

3.4 Reward Shaping

Most of the reward functions in dialogue scenarios are usually manually defined. Typically, a -1 for each turn and a significant positive or negative reward indicating the status of the dialogue at the end of a session. Such sparse reward is one of the reasons that RL agents have poor learning efficiency. Initially, the agents are fain to explore state-action uniformly at random. To this end, we propose to use reward shaping to integrate priors into RL learning to alleviate reward sparsity.

Reward shaping is a popular method to integrate prior knowledge into reward function to improve policy exploration (Brys et al., 2015). It provides the learning agent with an extra intermediate and

task-related reward that enriches the original reward signal:

$$r'(s,a) = r(s,a) + F_D(\cdot) \quad (6)$$

Where F_D denotes rewards from demonstrations. However, modifying the reward function may change the original MDPs and make the agent converge to a suboptimal point. (Wiewiora et al., 2003) proved that the MDP keeps unchanged and maintains convergency property if $F_D(\cdot)$ is defined as:

$$F_D(s,a,s',a') = \gamma\phi_D(s',a') - \phi_D(s,a) \quad (7)$$

where $\phi_D(s,a)$ is a potential function of state-action pair. Its definition is intuitive. We bonus these policy paths that were consistent with the demonstrations. As such, the value of $\phi_D(s,a)$ is expected to be high when action a is demonstrated in a state s^d similar to s, and if s is completely different from s_a^d, $\phi_D(s,a)$ should be close to 0. To achieve this, multi-variate Gaussian is used to compute the similarity between state-action pairs.

$$G(s,a,s^d,a^d) = \begin{cases} e^{(-\frac{1}{2}(s-s_a^d)^T\Sigma^{-1}(s-s_a^d))}, a = a^d \\ 0 \quad otherwise \end{cases} \quad (8)$$

We search through the demonstrations to obtain the sample with highest similarity:

$$\phi_D(s,a) = \max_{s_a^d} G(s, s_a^d) \quad (9)$$

Using reward shaping to learn policy has several advantages. It leverages demonstrations to bonus these state-actions that are similar to demonstrations. The reward calculated from reward shaping is more informative and demonstration guided than the human-defined reward, which mitigates the reward sparsity issue to some degree.

4 Experiments and Results

We evaluate the proposed S^2*Agent* with a user simulator on several public task-oriented datasets, including movie ticket booking, restaurant reservation, and taxi reservation. Additionally, to asses the generalization capability of shaping mechanism, We conduct domain adaptation experiments. Finally, human evaluation results are reported.

4.1 Dataset

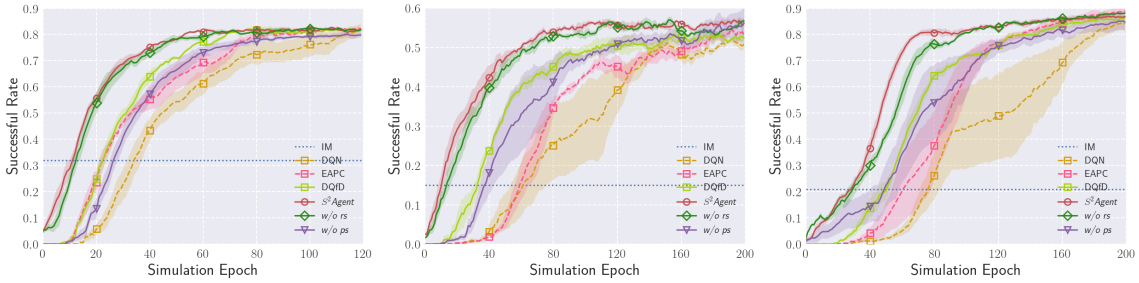
The raw conversation data in the movie ticket booking task are collected through Amazon Mechanical Turk, and the data for the restaurant reservation and taxi calling scenario is provided by Microsoft Dialogue Challenge [4]. The three datasets have been manually labeled based on a schema defined by domain experts. We extend and annotated movie booking task with a payment scenario to simulate the situation of extending the dialogue system with new slots and values. All datasets contain 11 intents. The movie dataset contains 13 slots, and the other three contain 29 slots. Detailed information about the intents and slots is provided in Appendix A table 3.

4.2 Baseline Agents

To benchmark the performance of the shaping mechanism, we have developed different versions of task-completion dialogue agents for comparison as follows:

- **Imitation Model (IM)** agent is implemented with Multi-Layer Perception and trained with the human demonstrations data to predict actions given dialogue states.
- **DQN** agent is learned with Deep Q-Network.
- **EAPC** Teaching via Example Action with Predicted Critique (EAPC) introduced in (Chen et al., 2017a) leverages real-time human demonstrations to improve policy learning. EAPC assumes the existence of human teachers during the learning process. It receives example actions from human teachers and, in the meantime, trains an action prediction model with the example actions as a critic for turn-level reward shaping. Since human teachers are not available in our case, we implement EAPC in the absence of teachers but use the same amount of human demonstrations to train a weak action prediction model. If the predicted action is identical to the action given by the policy model, the agent receives an extra positive reward otherwise an extra negative reward. This method can be viewed as a variant of S^2*Agent* with only reward shaping using noise reward estimations from the imitation model.
- **DQfD** (Hester et al., 2018) agent also leverages human demonstrations to improve policy learning. It adds additional classification

[4] https://github.com/xiul-msr/e2e_dialog_challenge



(a) Movie (b) Restaurant (c) Taxi

Figure 2: Learning curves of all the agents in Movie, Restaurant and Taxi domains. All the agents use the same amount of human demonstrations.

Table 1: The performance of the average turn and average reward of different agents in different domains. $w/o\ rs$ denotes S^2Agent without reward shaping; $w/o\ ps$ denotes S^2Agent without policy shaping;* denotes significant level $p < 0.05$ with other baselines except DQfD in movie domain. Succ. denotes success rate.

Agent	Movie			Restaurant			Taxi			Movie-Ext		
	Succ.↑	Turn↓	Reward↑	Succ.↑	Turn↓	Reward↑	Succ.↑	Turn↓	Reward↑	Succ.↑	Turn↓	Reward↑
IM	0.33	32.62	-11.47	0.16	**37.56**	-52.03	0.22	**15.07**	-27.33	0.37	**35.38**	-8.84
DQN	0.82	37.13	21.57	0.51	50.84	-31.65	0.84	45.69	-10.08	0.66	57.01	-40.37
EAPC	0.82	30.66	42.22	0.53	48.07	-24.86	**0.88**	38.71	19.10	0.65	55.34	-31.50
DQfD	0.81	27.53	50.57	0.52	43.90	-5.10	0.86	34.88	32.33	0.61	53.64	-19.34
*S^2Agent **	**0.82**	**21.25**	**72.68**	**0.57**	38.35	**16.40**	0.87	27.25	**61.50**	**0.70**	49.97	**3.39**
w/o rs	0.82	23.30	69.20	0.57	39.66	11.85	**0.88**	28.14	55.47	0.67	51.03	-4.93
w/o ps	0.80	31.68	40.28	0.57	45.40	-9.45	0.85	35.39	28.20	0.65	53.68	-19.27

loss from human demonstrations to DQN to ensure that the agent predicts correct actions on human demonstrated states. In the early learning phase, DQfD is trained only with the demonstrations to obtain a policy that mimics the human. Then, accumulated experiences mixed with the demonstration are used to train DQfD.

- S^2***Agent*** is our proposed agent that is trained with both policy shaping and reward shaping, as described in Algorithm 1.
- S^2***Agent*** **w/o rs** is a variant of S^2Agent which learns policy with only policy shaping to reconcile the final action.
- S^2***Agent*** **w/o ps** is a variant of S^2Agent but only has reward shaping to bonus state-actions similar to demonstrations.

Implementation Details Imitation model agents for all domains are single layer MLPs with 50 hidden dimensions and $tanh$ as the activation function. The IM agent is also used in policy shaping to reconcile the policy. All RL-based agents (DQN, DQfD, S^2Agent) are MLPs with $tanh$ activations. Each policy network Q(.) has one hidden layer with 60 hidden nodes. All the agents are trained with the same set of hyper-parameters. ϵ-greedy is utilized for policy exploration. We set the discount factor as $\gamma = 0.9$. The target network is updated at the end of each epoch. To mitigate warm-up issues, We build a naive but occasionally successful rule-based agent to provide experiences in the beginning. For a fair comparison, we pre-fill the experience replay buffer D^a with human demonstrations for all the variants of agents (Lipton et al., 2016). Confidence factor C used in policy shaping is set 0.7. As for the reward shaping, γ in equ.7 is set as 1.

4.3 User Simulator

Training RL-based dialogue agents require an environment to interact with, and it usually needs a large volume of interactions to achieve good performance, which is not affordable in reality. It is commonly acceptable to employ a user simulator to train RL-based agents (Jain et al., 2018; Li et al., 2016; Schatzmann et al., 2007).

We adopt a public available agenda-based user simulator (Li et al., 2016) for our experiment setup. During training, the simulator provides the agent with responses and rewards. The reward is defined as -1 for each turn to encourage short turns and a

(a) Movie (b) Restaurant (c) Taxi

Figure 3: The effect of number of human demonstration on the performance. The moving averaged success rate is calculated within 120 epochs for Movie, 200 epochs for Restaurant, and 200 epochs for Taxi.

large positive reward ($2L$) for successful dialogue or a negative reward of L for failed one, where L (set as 70) is the maximum number of turns in each dialogue. A dialogue is considered successful only if the agent helps the user simulator accomplish the goal and satisfies all the user's search constraints. In addition, the average number of turns and the average reward are also reported to evaluate each model.

4.4 Simulator Evaluation

Main Results. The main simulation results are shown in Table 1 and Figure.2, 3, 4. The results show that with shaping mechanisms, S^2*Agent* learns much faster and performs consistently better than DQN and DQfD in all the domains with a statistically significant margin.

Figure 2 shows the learning curve of different agents in different domains. Firstly, the DQN agent performs better than the IM agent, which is not surprising since it interacts with the simulator and is optimized to solve user goals. DQfD and EAPC agents leverage human demonstrations to mitigate the reward sparsity issues. Their performances are consistently better than DQN. Besides, S^2*Agent w/o ps* uses reward shaping to alleviate reward sparsity by bonusing additional rewards for states that are consistent with demonstrations. As a consequence, it performs better than DQN in all the domains. Though EAPC has a similar reward shaping mechanism, its reward estimation relies heavily on the qualify of the action prediction model. As such, EAPC performs slightly worse than S^2*Agent w/o ps*. In addition, policy shaping reconciles the agent action with knowledge learned from human demonstrations. It biases the agent to explore these actions which human expert does. As shown in figure 2, S^2*Agent w/o rs* learn the dialogue policy much faster than all the baselines. In the Movie domain, it achieves nearly a 60% success rate using only 20 epochs. By contrast, the second-best agent DQfD only achieves a 20% successful rate at epoch 20. Similar results are also observed in Restaurant and Taxi domains. When integrating both policy shaping and reward shaping to DQN, S^2*Agent* achieves the best performance and is more data-efficient. For example, S^2*Agent* in the Taxi domain achieves approximately 60% successful rate at 50 epoch while the following competitor only has around 40% successful rate. The above observation also confirms that policy shaping and reward shaping operate in different dimensions, which means policy shaping improves the learning by directly calibrating in the action space and reward shaping in the value function space, and are mutual-complementary. Noted that the improvement of combining policy shaping and reward shaping in the Movie domain is not as significant as that in Restaurant and Taxi. This is too large degree attributed to the increased complexity of Restaurant and Taxi dataset, which have two times more slots than the Movie dataset, meaning that the state-action space is much larger than the movie domain and posing more challenges in exploration. Under this situation, policy shaping and reward shaping benefit the S^2*Agent* to a large extent.

Results of training with varying number of demonstrations. Intuitively, the number of human demonstrations has a large impact on policy learning. The imitation model agent might be able to summarize a good expert policy when a large volume of human demonstrations is available. However, we hope the shaping mechanism is capable of improving learning efficiency with limited human demonstrations for RL-base agents. As such, we

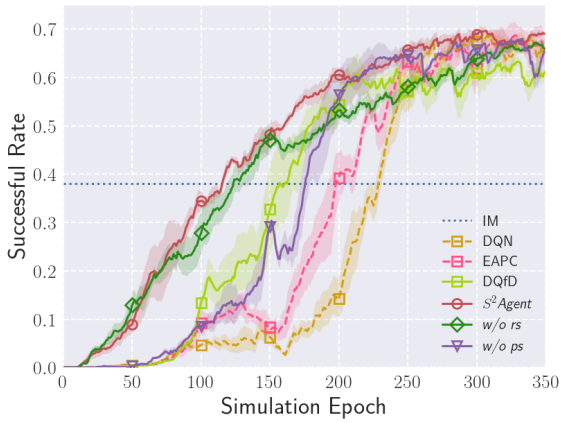


Figure 4: Learning curves of different agents in Movie-Ext domain, all the agents are adapted from trained agents in Movie domain.

experiment with different sizes of demonstrations between 25 and 125 to asses the effect of different numbers of human demonstration on learning efficiency and quality. Figure 3 shows the average performance of each agent during learning, which indicates the learning speed and quality. Our proposed shaping mechanisms improve policy learning speed and quality and are robust to the number of demonstrations. Even with the small number of human demonstrations as 25, S^2*Agent* achieves a 5% higher success rate than DQfD and EAPC in the Movie domain and 10% in the Taxi domain. As the number of demonstrations increases, the gap between DQfD and S^2*Agent* becomes larger, showing that policy mechanisms can still benefit from more human demonstrations available.

Results of domain extension Typically, RL-based agents are built with a fixed ontology. However, a dialogue system should be able to evolve as being used to handle new intents, slots, unanticipated actions from users. To asses the ability of quickly adapting to the new environment, we extend existing movie user simulator, denoted as Movie-Ext, to simulate domain adaption scenario. Movie-Ext has an additional payment task requiring the agent to converse with users to firstly book a ticket and then finish the payment. Details about the extended intent/slots can be found in the in appendix Table.3. All the agents are continually optimized from the previously trained agents for the movie ticket booking task. Meanwhile, we additionally collect a small number of human demonstrations to update the IM agent. Figure 4 shows the learning curves of different agents on the extended task. As we can see, both S^2*Agent* and S^2*Agent* *w/o rs* can quickly adapt to the new environment and outperform the IM agent, with only 150 epochs it achieves around 50% success rate. Though DQfD explicitly leverages human demonstrations, it still lags behind *w/o rs*, showing that shaping in the policy space is more effective than solely adding supervised learning loss for Q-learning. Reward shaping also benefits DQN to explore better policy. These observations confirm that S^2*Agent* with shaping mechanism is capable of quickly adapting to the new environment.

4.5 Human Evaluation

User simulators are not necessary to reflect the complexity of human users (Dhingra et al., 2017). To further evaluate the feasibility of S^2*Agent* in real scenarios, We deploy the agents in Table 1 to interact with real human users in Movie and Movie-Ext domains [5].

Table 2: Human evaluation results on Movie and Movie-Ext domains. We use models at epoch 50 and epoch 200 for Movie domain and Movie-Ext, respectively. $w/o\ rs$ denotes S^2*Agent* without reward shaping; $w/o\ ps$ denotes S^2*Agent* without policy shaping; * denotes significant level $p < 0.05$ with other agents. Succ. denotes success rate.

Model	Movie		Movie-Ext	
	Succ.↑	Rating↑	Succ.↑	Rating↑
IM	0.42	3.92	0.40	1.96
DQN	0.56	3.36	0.26	2.68
EAPC	0.68	3.96	0.34	3.12
DQfD	0.72	3.92	0.50	3.24
S^2*Agent* *	**0.74**	**4.36**	**0.62**	**3.56**
w/o rs	0.72	4.26	0.46	2.94
w/o ps	0.70	4.12	0.52	3.20

All evaluated agents are trained with 50 epochs and 200 epochs for Movie and Movie-Ext respectively. In each dialogue session, one of the agents is randomly selected to converse with a human user. Each user is assigned with a goal sampled from the corpus and is instructed to converse with the agent to complete the task. Users have the choice of terminating the task and ending the session at any time if users believe that the dialogue is unlikely to succeed or simply because the agent repeats for several turns. In such a case, the session is considered as a failure. Finally, at the end of each session, users are required to give explicit feedback on whether the dialogue succeeded (i.e., whether

[5]For the time and cost consideration, we only conduct experiments on Movie and Movie-Ext domains.

the movie tickets were booked (and paid) with all the user constraints satisfied). Additionally, users are requested to rate the session on a scale from 1 to 5 about the quality/naturalness (5 is the best, 1 is the worst). We collect 50 dialogue sessions for each agent. The results are listed in Table 2. *S^2Agent* and *S^2Agent w/o rs* perform consistently better than DQN and DQfD, which is consistent with what we have observed in simulation evaluation. In addition, *S^2Agent* achieves the best performance in terms of success rate and user rating.

5 Conclusion

In this paper, we present a new strategy for learning dialogue policy with human demonstrations. Compared with previous work, our proposed *S^2Agent* is capable of learning in a more efficient manner. By using policy shaping and reward shaping, *S^2Agent* can leverage knowledge distilled from the demonstrations to calibrate actions from underlying RL agents for better trajectories, and obtains extra rewards for these state-actions similar to demonstrations alleviating reward sparsity for better exploration. The results of simulation and human evaluation show that our proposed agent is efficient and effective in both single domain and a challenging domain adaptation setting.

Acknowledgments

We appreciate the efforts from the anonymous reviewers; they have helped us improve this paper a lot. The research described in this paper is partially supported by Hong Kong RGC-GRF grant 14204118.

References

Tim Brys, Anna Harutyunyan, Halit Bener Suay, Sonia Chernova, Matthew E Taylor, and Ann Nowé. 2015. Reinforcement learning from demonstration through shaping. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Thomas Cederborg, Ishaan Grover, Charles L Isbell, and Andrea L Thomaz. 2015. Policy shaping with human teachers. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Lu Chen, Runzhe Yang, Cheng Chang, Zihao Ye, Xiang Zhou, and Kai Yu. 2017a. On-line dialogue policy learning with companion teaching. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 198–204.

Lu Chen, Xiang Zhou, Cheng Chang, Runzhe Yang, and Kai Yu. 2017b. Agent-aware dropout dqn for safe and efficient on-line dialogue policy learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2454–2464.

Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2017. Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 484–495.

Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. 2013. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in neural information processing systems*, pages 2625–2633.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. 2018. Deep q-learning from demonstrations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Alankar Jain, Florian Pecune, Yoichi Matsuyama, and Justine Cassell. 2018. A user simulator architecture for socially-aware conversational agents. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pages 133–140. ACM.

Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Zheng Zhang, Yaoqin Zhang, Xiang Li, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, and Jianfeng Gao. 2019. ConvLab: Multi-domain end-to-end dialog system platform. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 64–69, Florence, Italy. Association for Computational Linguistics.

Jinchao Li, Baolin Peng, Sungjin Lee, Jianfeng Gao, Ryuichi Takanobu, Qi Zhu, Minlie Huang, Hannes Schulz, Adam Atkinson, and Mahmoud Adada. 2020. Results of the multi-domain task-completion dialog challenge. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, Eighth Dialog System Technology Challenge Workshop*.

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*.

Zachary Lipton, Xiujun Li, Jianfeng Gao, Lihong Li, Faisal Ahmed, and Li Deng. 2018. Bbq-networks:

Efficient exploration in deep reinforcement learning for task-oriented dialogue systems. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zachary C Lipton, Jianfeng Gao, Lihong Li, Xiujun Li, Faisal Ahmed, and Li Deng. 2016. Efficient exploration for dialog policy learning with deep bbq networks & replay buffer spiking. *CoRR abs/1608.05081*.

Bhaskara Marthi. 2007. Automatic shaping and decomposition of reward functions. In *Proceedings of the 24th International Conference on Machine learning*, pages 601–608. ACM.

Dipendra Misra, Ming-Wei Chang, Xiaodong He, and Wen-tau Yih. 2018. Policy shaping and generalized update equations for semantic parsing from denotations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2442–2452, Brussels, Belgium. Association for Computational Linguistics.

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.

Shakir Mohamed and Danilo Jimenez Rezende. 2015. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2125–2133.

Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, Yun-Nung Chen, and Kam-Fai Wong. 2018a. Adversarial advantage actor-critic model for task-completion dialogue policy learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6149–6153. IEEE.

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

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. *arXiv preprint arXiv:1704.03084*.

Olivier Pietquin, Matthieu Geist, Senthilkumar Chandramohan, and Hervé Frezza-Buet. 2011. Sample-efficient batch reinforcement learning for dialogue management optimization. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(3):7.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152. Association for Computational Linguistics.

Pei-Hao Su, Pawel Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. *arXiv preprint arXiv:1707.00130*.

Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Continuously learning neural dialogue management. *arXiv preprint arXiv:1606.02689*.

Pei-Hao Su, David Vandyke, Milica Gasic, Nikola Mrksic, Tsung-Hsien Wen, and Steve Young. 2015. Reward shaping with recurrent neural networks for speeding up on-line policy learning in spoken dialogue systems. *arXiv preprint arXiv:1508.03391*.

Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 100–110, Hong Kong, China. Association for Computational Linguistics.

Eric Wiewiora, Garrison W Cottrell, and Charles Elkan. 2003. Principled methods for advising reinforcement learning agents. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 792–799.

Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *arXiv preprint arXiv:1702.03274*.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Table 3: The data annotation schema.

	Movie	Restaurant	Taxi	Movie-Ext
Slots	city, numberofpeople, theater, zip, distanceconstraints, theater chain, video format, state, starttime, date, moviename, ticket, taskcomplete	city, closing, date, distanceconstraints, cuisine, greeting, restaurantname, numberofpeople, numberofkids, taskcomplete, other, pricing, starttime, state, zip, address, reservation, theater, atmosphere, rating, dress code, food, mealtype, choice, seating, occasion, personfullname, phonenumber, restauranttype	car type, city, closing, car level, date, distanceconstraints, dropoff location, greeting, name, driver id, numberofpeople, other, pickup location, dropoff location city, budget, pickup location city, pickup time, speed, state, cost, taxi company, mc list, taskcomplete, taxi, zip, result, driver level, numberofkids, emergency degree	city, numberofpeople, theater,zip, distanceconstraints, theater chain, video format, state, starttime, date, moviename, ticket, taskcomplete, **bill, cost, tax, bill number, bank, service fee, pay type, discount, consumption point, credit card point**
Intent	request, inform ,confirm question, confirm answer, greeting, closing, multiple choice, thanks, welcome, deny, not sure			

Table 4: The performance of Imitation Model on different dataset.

Domain	#Pair	Precision	Recall	F1-score
Movie	50	0.76	0.86	0.81
Restaurant	50	0.73	0.80	0.76
Taxi	50	0.83	0.90	0.86
Movie-Ext	100	0.84	0.83	0.82

A Appendices

Table 3 lists all annotated dialogue acts and slots in details. Table 4 lists the training results of Imitation Model on all dataset.