High-Quality Diversification for Task-Oriented Dialogue Systems

Zhiwen Tang **Hrishikesh Kulkarni** **Grace Hui Yang**
InfoSense, Department of Computer Science
Georgetown University
{zt79,hpk8,grace.yang}@georgetown.edu

Abstract

Many task-oriented dialogue systems use deep reinforcement learning (DRL) to learn policies that respond to the user appropriately and complete the tasks successfully. Training DRL agents with diverse dialogue trajectories prepare them well for rare user requests and unseen situations. One effective diversification method is to let the agent interact with a diverse set of learned user models. However, trajectories created by these artificial user models may contain generation errors, which can quickly propagate into the agent's policy. It is thus important to control the quality of the diversification and resist the noise. In this paper, we propose a novel dialogue diversification method for task-oriented dialogue systems trained in simulators. Our method, Intermittent Short Extension Ensemble (I-SEE),[1] constrains the intensity to interact with an ensemble of diverse user models and effectively controls the quality of the diversification. Evaluations on the Multiwoz dataset show that I-SEE successfully boosts the performance of several state-of-the-art DRL dialogue agents.

1 Introduction

Task-oriented dialogue agents assist human users to complete their tasks in multi-round human-agent interactions. Example tasks include booking a movie ticket or reserving a lunch table. Many agents use deep reinforcement learning (DRL) to learn good policies that respond appropriately in the dialogue and succeed in completing the task (Schulman et al., 2017; Takanobu et al., 2020; Peng et al., 2018). Due to the high cost of interactive training using real human users, simulators have been used to replace the humans (Shi et al., 2019). Early user simulators rely on rules hand-crafted by domain experts, who review and summarize dialogue templates from everyday conversations (Li et al., 2016b). Rule-based simulators manage routine task scenarios efficiently; but they cannot easily generate unconventional and more diverse dialogue acts marked by human spontaneity. Agents learning from these simulators may fail to handle rare queries and unseen situations. For example, in Table 1's left column, the system learns from a rule-based simulator and fails to answer an unexpected question "*In which area of the city is this hotel located?*" The user immediately ends the conversation and does not book the hotel.

Figure 1: Agent Learns from Diversified Simulators.

Increasing dialogue diversity is a long-lasting research interest motivated by different needs. In dialogue generation, diversity is desired to avoid dull responses. In simulator construction and reinforcement learning (RL), diversity is desired for training robust agents. Although the uses are slightly different, ideas to improve diversification can be universal. Dialogue diversity can be improved via i) enforcing diversity in objective functions (such as maximize mutual information) of neural models (Li et al., 2016a; Baheti et al., 2018), ii) perturbing language rules (Niu and Bansal, 2019) or environment parameters (Tobin et al., 2017; Ruiz et al., 2019), iii) randomizing trajectory synthesis (Andrychowicz et al., 2017; Lu et al., 2019), iv) selecting more diverse data contributors (Stasaski et al., 2020), and v) sampling trajectories from a diverse set of environments (Chua et al., 2018; Janner et al., 2019).

Figure 1 illustrates the last idea, which has received much attention recently (Chua et al., 2018; Janner et al., 2019). This line of research diversi-

[1] I-SEE codes can be found at https://github.com/smt-HS/I-SEE.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1861–1872
August 1–6, 2021. ©2021 Association for Computational Linguistics

No Diversification	Full Diversification	Quality-Controlled Diversifica. (I-SEE)
User: Hi! I would like to book hotel Leverton House. System: Leverton House is a cheap hotel. User: What is the post of Leverton House? System: Post of Leverton House is cb58rs. User: What is the type of the hotel? System: It is a guesthouse. User: Where is it located? System: Leverton House is located at 732-734 new market road. User: What is the hotel phone number? System: You can call the hotel at 01223292094. User: In which area of the city is this hotel located? System: There are many attractions in the centre of the city. [irrelevant] User: Ok, bye! System: Goodbye!	User: Hi! I would like to book hotel Leverton House. System: Hotel Leverton House is located on the east side of the city. User: What is the post of the hotel? System: The post of hotel Leverton House is cb58rs. ... System: ... Shall I go ahead and book it? User: Ok! I am not sure about the duration of stay or the number of people staying. System: Booking confirmed. Anything else? [success] User: I would also like to book a train from Peterborough to Cambridge arriving this Sunday at 15:15. System: Found a train arriving at 8:38. [incorrect] User: I want the one arriving at 15:15. System: Found a train arriving at 6:09. [incorrect] User: Ok, Bye! System: Have a good day!	User: Hi! I would like to book hotel Leverton House. System: Are you interested in booking hotel Leverton House on the east side? User: Where is it exactly located? What is the post, phone and type of the hotel? ... System: ... Would you like to go ahead and book it? User: Ok, Not sure on the duration of stay or number of people as of now. System: Booking confirmed. [success] User: I would also like to book a train from Peterborough to Cambridge arriving at 15:15 on Sunday. System: Would you like to go ahead and book tickets on TR7721? [correct] User: Ok. Not sure on the number of people. System: One ticket booked. [success] User: Great! Thanks! System: Have a good day!
Failure: The system fails to answer the area question from the user and it does not book the hotel for the user.	**Failure**: Even though the system successfully answers all questions from the user and books the hotel, it fails to identify the correct train based on arrival time and does not book the train ticket.	**Success**: Both hotel and train ticket have been successfully booked satisfying all constraints.

Table 1: Example Dialogues.

fies an agent's learning experiences by letting the agent interact with a diverse set of generative user models learned from an expert simulator. The idea involves little manual configuration and is often developed as model-based DRL (MBDRL) (Sutton and Barto, 1998). MBDRL methods alternate between learning an environment model and learning a policy. For a task-oriented dialogue agent, the environment model can be thought of a user model. It is a dynamic model updated to fit the trajectories the agent has collected so far; the policy then is optimized to maximize the expected long-term rewards within the model. Diversification of the user model is achieved by randomizing the parameter initialization of neural networks to our advantage. The agent, which is the policy learner, interacts with an ensemble of randomized user models to gain more diverse learning experiences.

However, one issue in this approach is that errors in (user) model learning may quickly propagate into policy learning. Table 1's middle column demonstrates a result from uncontrolled use of the diversified user models. In this example, even though the system successfully answers all questions from the user and books the hotel, the agent recommends two erroneous trains that do not satisfy the user's constraints and fail to book the ticket. This is because noise has been introduced to the training dialogues and they deviate too much from a legitimate conversation in real-life.

In this paper, we propose a novel dialogue diversification method, Intermittent Short Extension Ensemble (I-SEE), for task-oriented dialogues agents trained in simulators. First, I-SEE employs neural networks to learn a generative user model by imitating the expert simulator (Torabi et al., 2018). Second, it randomizes the parameter initialization of the neural networks to generate more user models, which are diversification from the original expert-built simulator. These randomized user models form an ensemble of diverse simulators, named Diverse User Model Ensemble (DUME). Third, during policy learning, the agent interacts with multiple simulators to obtain diverse training trajectories. Particularly, we propose to mix trajectory segments sampled from the expert simulator and trajectory segments sampled from the DUME. This is to constrain the degree of noise introduced by diversification and do not divert too far from the expert simulator. Moreover, we propose to include the DUME trajectories only moderately frequently and for a short horizon.

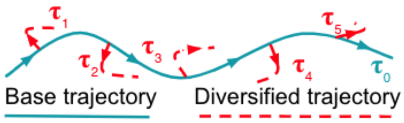


Figure 2: Conceptual illustration of I-SEE.

Figure 2 illustrates our idea conceptually. By constraining the degree of diversification, I-SEE effectively controls the training trajectories' quality while preserving their diversity. In Table 1's last (right) example, the I-SEE agent successfully takes the booking task to a logical conclusion by correctly finding the TR7721 train, which satisfies the user's time constraints. We apply I-SEE to a few best performing DRL dialogue methods and evaluate them on the Multiwoz (Budzianowski et al., 2018) dataset. Results show that using DUME and I-SEE in combination would significantly improve the performance of these state-of-the-art systems.

2 Related Work

2.1 Task-Oriented Dialogue Systems

Popular approaches for task-oriented dialogue systems include Sequence-to-Sequence (Seq2Seq) response generation (Vinyals and Le, 2015; Hosseini-Asl et al., 2020), knowledge graph-driven question answering (KG-QA) (Christmann et al., 2019; Moon et al., 2019; Young et al., 2018; Madotto et al., 2018, 2020), context-sensitive response retrieval (Aliannejadi et al., 2019; Yu et al., 2020; Qu et al., 2020; Wang and Ai, 2021) and RL (Buck et al., 2018; Peng et al., 2018; Tang and Yang, 2020; Luo et al., 2014; Li et al., 2017; Su et al., 2018; Peng et al., 2018).

Seq2Seq dialogue agents are generation methods. They use language models to capture the probability of one utterance given the previous, and based on the learned models to generate new utterances (Vinyals and Le, 2015; Hosseini-Asl et al., 2020). These supervised methods take advantage of deep neural networks and infer effective encoder-and-decoders from large amount of sequential training data. Modeling the dialogue states (Campagna et al., 2020) in the Seq2Seq architecture is a major interest in this line of research.

KG-QA dialogue agents enable reasoning and inference with pre-built knowledge graphs (KGs). The KGs can be about commonsense or domain-specific knowledge. A general KG can help a conversation more interesting and engaging (Moon et al., 2019; Young et al., 2018); while a specific KG can help accomplish the task more efficiently (Madotto et al., 2018, 2020). Methods in this category focus on scaling up the KGs (Madotto et al., 2020) and hopping mulitple steps on the KGs (Moon et al., 2019).

Retrieval-based dialogue agents leverage mature techniques in ad hoc retrieval and extend the techniques from individual queries to a session of them. Retrieval-based approaches do not rely on simulators; instead, learning from historical data, such as query logs, is still quite popular. This line of research focuses on revealing a user's mixed-initiative information need via asking back-and-force questions (Aliannejadi et al., 2019; Yu et al., 2020; Qu et al., 2020; Wang and Ai, 2021). However, when task complexity goes beyond the user's capability, these approaches may face difficulty in finding global solutions to the task goal.

RL-based dialogue agents can be grouped into model-free and model-based methods. Model-Free DRL (MFDRL) agents take a pre-built environment/simulator as it is and learn policies via direct interactions with it (Li et al., 2017; Dhingra et al., 2017; Li et al., 2017; Lipton et al., 2018; Su et al., 2018; Wu et al., 2020). On the contrary, model-based DRL (MBDRL) agents indirectly learn policies from the environment. MBDRL has two concurrent learning modules, namely model learning and policy learning. The *model learning* module can be thought of an additional computational layer between the environment and the agent. This provides opportunities to alter the original environment. MBDRL was originally proposed in robotics and control to speed up direct policy learning by inferring decision rules from past interactions and embedding them in the model. For dialogue agents, this middle layer of model learning acts as derived simulators (or learned user models) from the original expert simulator. Deep Dyna-Q (DDQ) (Peng et al., 2018) is an MBDRL method built upon Dyna (Sutton and Barto, 1998). D3Q (Su et al., 2018) employs generative adversarial networks (GAN) to minimize the difference between trajectories generated from the learned models and that from the original expert simulator, assuming that the expert simulator is the gold standard. Likewise, ADC (Wu et al., 2020) uses double critics to mitigate the impact of poorly-generated trajectories to stabilize the agent's performance. Our method belongs to the family of MBDRL, with a focus on diversification.

2.2 Diversification in Dialogues

Increasing dialogue diversity is a long-lasting research interest. Dialogue diversity can be improved via enforcing diversity objective functions (such as maximize mutual information) in neural models (Li et al., 2016a; Baheti et al., 2018), perturbing language rules (Niu and Bansal, 2019) or environment parameters (Tobin et al., 2017; Ruiz et al., 2019), randomizing trajectory synthesis (Andrychowicz et al., 2017; Lu et al., 2019), selecting more diverse data contributors (Stasaski et al., 2020), and sampling trajectories from a diverse set of environments (Chua et al., 2018; Janner et al., 2019). For instance, Campagna et al. augmented dialogue data using domain-independent transition rules and domain-specific ontology (Campagna et al., 2020). Niu and Bansal synthesized more diverse dialogue trajectories by choosing semantic-preserving language perturbations via RL (Niu and Bansal, 2019).

2.3 Diversification in DRL

In model-free DRL, diversification can be achieved by domain randomization (Tobin et al., 2017; Ruiz et al., 2019) or hindsight experience replay (Andrychowicz et al., 2017; Lu et al., 2019), without modeling the dynamics of the environment.

In model-based DRL, diversification is done by altering the learned environment/user model; which are the closest to our work. For instance, Chua et al. proposed probabilistic ensemble trajectory sampling (PETS) (Chua et al., 2018), which learns an ensemble of environment models and uses them for planning. The follow-up work (Janner et al., 2019) extended PETS with policy learning. Like us, Janner et al. concerned noise added by new trajectories generated by the derived environments. They proposed that the generation of new trajectories from the derived models should start from a beginning state shared with the original environment. These methods are mainly developed for robotics and work in continuous action space.

In this paper, we propose to obtain mixed training trajectories by branching from the original trajectory generated by the expert simulator and extending with new trajectories by the derived simulators. Different from (Janner et al., 2019), our method is designed for dialogue agents' discrete action space. In our method, each training trajectory has an overlap much larger than (Janner et al., 2019) has with the expert trajectory. This allows us to obtain smoother transition distributions to facilitate discrete action space better. In addition, our method can parameterize the intensity to branch out, so that the level of diversification can be controlled and adjusted.

3 Problem Setup

Task-Oriented Dialogue is the interactive process between a user and a dialogue agent, who work together to accomplish a task. The process begins with the user initiating the dialogue with a task goal in mind. The task goal can have *constraints* and *requests*. Constraints are requirements a system response must satisfy and requests are for missing information the user needs to accomplish the task. E.g., a user wants to book tickets of a movie to be played on weekends but does not know the theater's phone number. Here the constraint is $time = weekend$ and request is $phone\_number =?$. The dialogue ends when both parties say "good-bye" or the user abandons it.

Expert Simulator is the rule-based simulator built by human experts. It is denoted as M_0, which describes how a typical user would choose proper dialogue acts as the dialogue unfolds. The state of the expert simulator is s_t^u at time step t and the action is a_t^u selected from an action set A^u, which can be either *making requests* or *imposing constraints*. M_0 shows a mapping from s_t^u to a_t^u, describing patterns and behaviours for the human users, and provides feedback to and converse with the dialogue agent.

Diversified Simulator (or Diversified User Model) M_{ϕ_*} is a trainable user model that learns a parametric mapping from s_t^u to ${a'}_t^u$ with parameter ϕ_*. It mimics the behavior of the expert simulator M_0. With different parameter initialization, we can create a set of diversified user models. This set of diversified simulators is called Diversified User Model Ensemble (**DUME**).

Dialogue Agent (DA) is the automatic response generator, who is expected to search in the knowledge base, reply the human users with relevant and correct answers, and make transactions following the user's requests. We use s_t^s, a_t^s to denote the state and action of the dialogue agent at time step t. The agent also receives a reward signal r_t as immediate feedback for its action a_t^s. Its state transition function P models the probability of its next state given the current state and actions from both the user and the DA: $s_{t+1}^s = P(s_t^s, a_t^s, a_t^u)$. In the DRL setting, the DA is the policy learner. It learns a policy π from a set of dialogue trajectories $\{\tau_*\}$. The goal of the agent is to learn a policy that can maximize the expected cumulative rewards $\mathbb{E}_\pi[\sum_t r_t]$ in a task-oriented dialogue.

Interaction Tuple $\mathscr{T}$ is the state-action-reward tuple generated when the DA interacts with a simulator or a real user. At the tth dialogue turn, the tth interaction tuple is $\mathscr{T}_t = (s^s_t, a^s_t, r_t, s^u_t, a^u_t)$.

Trajectory Segment τ^k_j is a sequence of interaction tuples when the DA interacts with a simulator (M_0 or M_{ϕ_i}) or a real user, starting from time step j to k: $\tau^k_j = [\mathscr{T}_j, \mathscr{T}_{j+1+1}, ..., \mathscr{T}_k]$, where $\mathscr{T}_{t\in[j,k]}$ is the tth interaction tuple of the segment. Decided by the state transition function P, latter interaction tuples in τ depend on the earlier tuples. A **base trajectory segment** τ_0 is a trajectory that records the interaction between the expert simulator M_0 and the DA. A **diversified trajectory segment** τ' is a trajectory segment that records the interaction between a diversified simulator M_{ϕ_*} and the DA. A **full trajectory** τ^{full}_0=$[\mathscr{T}_0..., \mathscr{T}_T]$ starts from the beginning of a dialogue, s.t., $j = 0$ and ends at T, where T is the entire dialogue's length.

4 Proposed Work

Our method aims to provide high-quality diversified training trajectories for task-oriented dialogue agents. We propose to (1) construct an ensemble of diversified user models called DUME and (2) intermittently branching out short trajectories from the base trajectory using DUME and employ the new trajectories in policy learning.

Figure 3 illustrates the proposed system architecture. In our design, the dialogue agent can interact with both the expert simulator and a diversified simulator. Usually the agent starts with interacting with the expert simulator since $t = 0$. At a branching step $t = p$, the agent switches to the diversified simulator to interact with, until the trajectory ends at $t = T$. The diversified simulator is obtained via imitation learning (from the expert simulator) and neural network initialization randomization. By controlling how frequently the branching should be performed and how long a diversified segment should be used, we effectively reach a balance between training data diversity and quality.

4.1 Constructing Diversified User Model Ensemble (DUME)

To enhance dialogue diversity, we propose to have the agent interact with an ensemble of diverse user models $\{M_{\phi_*}\}$. We use neural networks with different initialization to learn diversified user models from the expert simulator M_0, and form the DUME using these learned models.

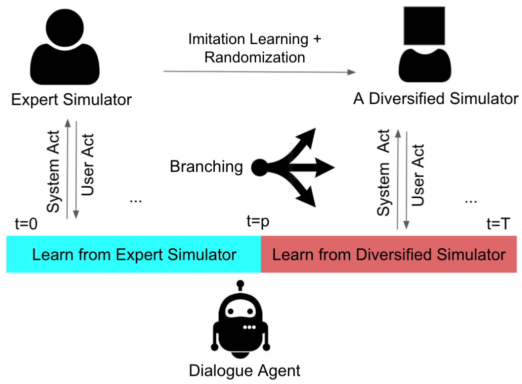


Figure 3: System Architecture

4.1.1 Learning a single user model

We propose to learn the user models from the expert simulator by behavior cloning (Torabi et al., 2018). For a single user model, we aim to learn a sequential decision-making function that maps $(s^u_1, s^u_2, ..., s^u_t...)$ to $(a^u_1, a^u_2, ..., a^u_t, ...)$. The training inputs are from the base trajectories τ_0, which includes a sequence of user state and user action pairs $\langle s^u_t, a^u_t\rangle$. The user state at the tth turn is

$$s^u_t = (G, \bigcup_{t'=1}^{t-1} a^s_{t'}) \tag{1}$$

where G is the user goal, which can include both constraints and requests. $\bigcup_{t'=1}^{t-1} a^s_{t'}$ is the history of the dialogue agent's actions. The user action a^u_t is

$$a^u_t = (a^u_{t,1}, a^u_{t,2}, ..., a^u_{t,|A^u|}) \tag{2}$$

where $a^u_{t,i}$ are binary variables indicating whether the ith dialogue act is active at dialogue turn t. A^u are the available dialogue acts for the user. The ending of a dialogue is also a special dialogue act.

Here a single user action can contain multiple dialogue acts. For instance, informing the destination and arrival time at the same dialogue turn when booking a train ticket. It means the number of dialogue acts per user action would vary. To allow the flexibility for modeling varied number of user acts, we propose to break the training trajectory (which is a sequence) τ_0 into individual state-action pairs and formulate the learning as choosing the right dialogue acts at a given state, i.e. learning the mapping from s^u_t to a^u_t. The optimization is done by minimizing the loss function $L(\phi)$:

$$L(\phi) = -\frac{1}{|A^u|}\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{|A^u|} a_{t,i}\log M_\phi(s_t^u)_i + (1-a_{t,i})\log(1-M_\phi(s_t^u)_i) \tag{3}$$

where ϕ is the model parameter vector, $a_{t,i}$ is the ground truth indicator of whether the ith dialog act is taken at time step t, and $M_\phi(s_t^u)_i$ estimates the probability of the ith dialog act being chosen by the user model given s_t^u. The learning is performed by a multi-layer perceptron neural network parameterized by ϕ.

We are aware that the learning of the user models can be done using much more sophisticated methods. E.g., we can use more advanced neural network architectures and/or incorporate more information when defining the user states. However, these changes are not the main focus of this paper. The proposed user modeling is sufficient to support our investigation in exploiting them to improve diversification.

4.1.2 Forming a Diverse Ensemble

We propose to build an ensemble of diversified user models for better diversification. The ensemble, DUME, contains a set of E number of user models $M_{\phi_1}, M_{\phi_2}, ..., M_{\phi_E}$. Each of them is trained with behavior cloning as stated in Section 4.1.1. DUME diversifies the user models by initializing the behavior cloning with different seeds. Each user model is trained using a separate neural network; these neural networks share the same architecture but use randomized, different initial parameters ϕ_j. Our experiments (Section 5.4) show that the diversity in DUME dramatically increases, as E increases.

4.2 Policy Learning with I-SEE

One would imagine that the more diversified trajectories used in training, the more robust the policy would be. An intuitive idea is to interact with the diversified user models M_{ϕ_*} from the beginning to the end, without using M_0 at all. DDQ (Peng et al., 2018) indeed exploits this design. However, a dialogue trajectory completely generated by M_{ϕ_*} suffers from accumulation of generation errors because they may deviate too much from what a real conversation looks like.

In this paper, we propose to learn from training trajectories generated from mixed sources. Our idea is to have controlled diversification during policy learning, where some of the learning is done

Algorithm 1: Trajectory Generation

Input : Simulator M,
Dialogue agent policy π,
Initial user state s_0^u,
Maximum trajectory length T

Output : Dialogue trajectory dataset D;

$D = \varnothing$;
Initialize the user state to s_0^u;
for T *time steps* **do**
  The user/simulator observes the state s_t^u and takes action $a_t^u = M^i(s_t^u)$;
  The agent observes the state s_t^s and takes action $a_t^s = \pi(s_t^s)$;
  The agent receives reward r_t;
  Store the interaction tuple $\langle s_t^s, a_t^s, r_t, s_t^u, a_t^u\rangle$ in D;
  if *the user/simulator decides to end the dialogue in* a_t^u **then**
    break;
  end
end
return D

by learning from the original expert simulator and some is done by learning from the diversified user models in DUME. The ratio of the diversified portion can be controlled as a hyper-parameter. The following details our method.

4.2.1 Diversifying the Trajectories

During policy learning, the dialogue agent collects training trajectories generated from the simulators, to keep refining its policy based on gradient ascent. Algorithm 1 details the trajectory generation process. In order to sample a trajectory, the policy learner, i.e. the dialogue agent, interacts with a user model to obtain interaction tuples step by step and store each individual tuple in a dataset D. To obtain an individual interaction tuple, the simulator needs to take an action based on its own user model, and then the agent performs an action based on the state and its current policy π. The agent receives rewards and the next state from the simulator. The interaction tuple is stored and would be used later to form a full trajectory. This process works the same regardless the agent interacting with the expert simulator or a diversified simulator.

In this work, we propose to diversify the agent's learning experiences by learning from trajectories generated from mixed sources. First, we generate a full base trajectory τ_0^{full}=[$\mathscr{T}_0$..., $\mathscr{T}_T$] from the expert simulator and store all its tuples. Second, we pick a branching tuple $\mathscr{T}_p \in \tau_0^{full}$ at a branching point $p \in (0, T)$. Third, from p onward, the trajectory is generated with a diversified user model M_{ϕ_*}, which would take an action a'^u_p different from the

Algorithm 2: Intermittent Short Extension Ensemble (I-SEE)

Input : Simulator ensemble size E
Branching horizon H
Diversification ratio η
Output : Dialogue agent's policy π

1 Initialize an ensemble of E user models;
2 Initialize the dialogue agent policy π;
3 **while** *the dialogue agent's policy does not converge* **do**
4 $D_{base}, D_{dvs} = \varnothing, \varnothing$;
5 **for** *every episode* **do**
6 Initialize the expert simulator M_0;
7 Observe the initial user state s_0^u;
8 D_{base} =TrajectoryGeneration(U, π, s_0^u, ∞);
9 **end**
10 **while** $|D_{dvs}| < \eta|D_{base}|$ **do**
11 Sample a simulator M_{ϕ_j} from the ensemble;
12 Sample a state s_t^u from D_{base} as the start state;
13 $D_{dvs} = D_{dvs} \bigcup$ TrajectoryGeneration$(M_{\phi_j}, \pi, s_t^u, H)$;
14 **end**
15 Update the dialogue agent's policy π with $D_{base} \bigcup D_{dvs}$;
16 Update the simulator ensemble with D_{base} using Eq. 3;
17 **end**

expert action a_p^u and the agent would also land in a different state ${s'}_{p+1}^s = P(s_p^s, a_p^s, {a'}_p^u)$.

Such interaction with the diversified simulator M_{ϕ_*} continues with H steps, resulting a diversified trajectory segment. The **diversified trajectory segment** $\tau_p'^{p+H}$ records the interaction between M_{ϕ_*} and the agent, extending the base trajectory τ_0 from a branching point p and running from $p+1$ onward. It is denoted as:

$$\tau_p'^{p+H} = [\mathscr{T}_p, \mathscr{T}'_{p+1}, ..., \mathscr{T}'_{p+H}],$$

where p is the branching point and $p > 0$, and H is τ_p''s horizon. The first interaction tuple in τ_p' is copied from the pth turn in τ_0, i.e., $\mathscr{T}'_p = \mathscr{T}_p$. The **full trajectory with diversification** is thus $\tau_p^{full} = [\mathscr{T}_0, ..., \mathscr{T}_p, \mathscr{T}'_{p+1}, ..., \mathscr{T}'_{p+H}]$.

Our method generates parts of a dialogue with the diversified simulator and the other parts using the expert simulator. Each training trajectory thus has overlaps with the expert trajectory, which obtains smoother transition distributions to facilitate the discrete action space that a dialogue agent has.

4.2.2 Intermittent, Short Extensions

Further, we control the quality of diversification by using the DUME conservatively – only use the DUME trajectories for a short horizon and intermittently – to avoid accumulating generation errors.

Branching Horizon. The hyper-parameter H is the branching horizon that controls how far a trajectory is generated from DUME. The larger the horizon H, the more diverse the resulting trajectory. Setting H too small may cause the policy to be myopic as actions take time to show effects; whereas setting it too large may result in accumulation of errors. Our experiments show that using a moderately small $H = 5$ is preferable. An analysis is reported in the experiment section.

Branching Intensity. Another factor that determines the degree of diversification is the intensity of branchings. Instead of branching at every single step, our method only intermittently forks a diversified trajectory uniformly. This is done by setting a diversification ratio η between the times the agent interacting with the expert simulator M_0 and with DUME. The diversification ratio η is calculated as:

$$\eta = \frac{count(\mathscr{T}_i', \forall i \in D_{dvs})}{count(\mathscr{T}_j, \forall j \in D_{base})} \tag{4}$$

where $\mathscr{T}_i'$ is a diversified interaction tuple stored in D_{dvs} and $\mathscr{T}_j$ is an interaction tuple stored in D_{base}. D_{base} and D_{dvs} are collections of individual interaction tuples obtained as Lines 4-14 in Algo. 2. A larger η means more diversified the agent's learning is. Algo. 2 shows the entire I-SEE algorithm.

5 Experiments

5.1 Experimental Setup

• **Dataset.** We evaluate the proposed approach on the Multiwoz (Budzianowski et al., 2018) dataset. Multiwoz is a large-scale benchmark dataset for task-oriented dialogue systems. It has seven task domains, including restaurant, hotel, attraction, taxi, train, hospital and police. One dialogue may involve multiple task domains, which is a good resemblance of how people converse in real life. Multiwoz provides 8,438 labelled dialogues, each dialogue of which is annotated by experts with a sequence of dialogue states and respective dialogue acts. Table 2 shows the dataset statistics. The expert simulator in Multiwoz starts a conversation and takes turns with a dialogue agent to dialogue. The simulator may request information from the agent or give the agent permission to do new bookings. At each turn, the simulator or the agent can perform one or more dialogue acts. The agent is

#Domains	#Dialogues	Total #Turns	Avg #Turns per dialogue
7	8,438	113,556	13.46
#Slots	#Values	Total DB Entries	Avg Entries per domain[2]
24	4,510	3,116	623

Table 2: Dataset Statistics (Multiwoz).

expected to 1) provide correct answers to requested information and 2) complete the booking, if asked.

• **Evaluation Metrics. Success** is our main metric, which is the success rate over all dialogue tasks tested. A task is successful if and only if 1) all the requested information is provided, and 2) all the booked entities match the user's requirements. **Inform F1** evaluates whether an agent provides the information requested by the user. It is calculated as $F1 = \frac{2Prec*Recall}{Prec+Recall}$, where $Prec$ and $Recall$ are the precision and recall of the information replied by the agent. **Match** evaluates whether the booked entities satisfy the user's requirement. It scores 1 if the correct entity is booked, otherwise 0. In the case of multiple bookings, the scores are averaged across all bookings. **#Turns** measures the number of turns a dialogue last regardless of its success. The less the turns, the better.

• **Baselines.** We compare the performance of a few top-performing DRL dialogue agents on the Multiwoz dataset with three settings. The settings are 1) the algorithm without diversification, 2) with full and uncontrolled diversification, and 3) with I-SEE. These baseline systems include state-of-the-art MFDRL and MBDRL methods and best performing DRL agents on Multiwoz. **DQN** (Deep Q-Network) (Mnih et al., 2015) is an off-policy MFDRL method, which approximates the value function of state-action pairs with a deep neural network and learns the function using experience replay. **PPO** (Proximal Policy Optimization) (Schulman et al., 2017) is an on-policy MFDRL algorithm, which optimizes a surrogate objective function which restricts the change of action distributions in a policy update. **GDPL** (Guided Dialogue Policy Learning) (Takanobu et al., 2019) is the best performer on Multiwoz. It uses inverse RL to reconstruct reward function and optimizes its policy with PPO. **DDQ** (Deep Dyna-Q) (Peng et al., 2018) is an MBDRL algorithm designed for task-oriented dialogue agents. DDQ generates complete trajectories from its environmental models, which is equivalent to our setting of DQN+full diversification. **MADPL** (Multi-Agent Dialogue Policy Learning) (Takanobu et al., 2020) is a multi-agent MFDRL method that trains the system and the user simulator simultaneously. It is also a leading performer on Multiwoz.

• **Implementation Details** We use Multiwoz's agenda-based simulator (Zhu et al., 2020) as the expert simulator. The DUME and policy networks and value networks in the baselines are learned using three-layer multi-layer perceptrons (MLPs). A learned user model has an input dimension of 230 and output of 67, with a hidden layer of 200 units. The DRL dialogue agents all use an the input layer of 553 units. PPO's policy network uses a hidden layer of 200 units and output of 166. PPO's value network has a hidden layer of 50 and output of 1. DQN also uses a hidden layer of 200 units and output of 166. The I-SEE dialogue agent is trained with a mix of expert simulator and diversified simulators as presented in the paper and tested with only the expert simulator.

[2]Five out of seven domains require querying the database.

5.2 Effectiveness

Table 3 presents the experiment results. The proposed method I-SEE outperforms the original algorithms and the full diversification variants for all baselines on the main metric, success, and the number of turns. The best performance is given by GDPL+I-SEE, with a success rate of 93.2 and only 7.32 dialogue turns on average. Moreover, the I-SEE variants perform the best on *Inform F1* for PPO and DQN, and on *Match* for PPO and GDPL. The improvements are large. These results suggest that diversification in general improves a DRL dialogue agent's effectiveness. However, full and uncontrolled diversification may worsen the performance; while a moderate level of diversification as we propose is a better choice.

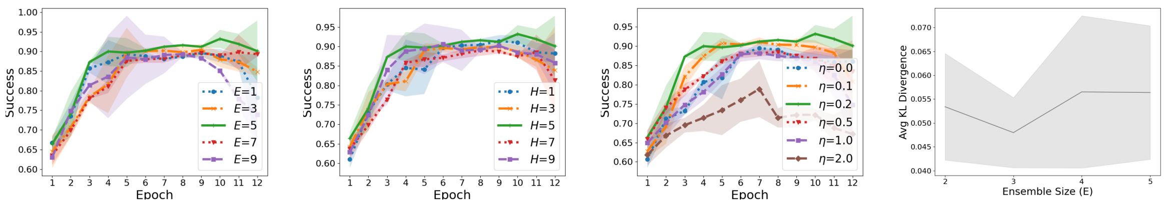
5.3 Analysis of I-SEE

To understand why I-SEE works, we investigate the relationship between the degree of diversification and the success rate. GDPL is selected as the baseline system X. We study three I-SEE hyper-parameters that are responsible for the degree of diversification. They are the user model ensemble size E, branching horizon H, and diversification ratio η. As each of these parameters gets bigger, the degree of diversification increases. We plot the dialogue agent's learning curves w.r.t the three parameters in Figures 4a, 4b, and 4c, respectively.

We observe that a single optimum exists for each hyper-parameter when they reach the best success

Algorithm	Success ↑	Impr.	Inform F1 ↑	Impr.	Match ↑	Impr.	#Turns ↓	Impr.%
MADPL	70.1		76.26		90.98		8.96	
PPO	77.9		86.45		78.90		9.785	
PPO+Dvs.	69.0	(-8.9)	80.27	(–6.18)	70.55	(-8.35)	11.39	(-16.40%)
PPO+I-SEE	**84.5**	**(+6.6, +15.5)**	**88.91**	**(+2.46, +8.64)**	**86.29**	**(+7.93, +15.74)**	**8.88**	**(+9.25%, +22.04%)**
DQN	74.4		87.61		**92.91** best		11.54	
DQN+Dvs. (DDQ)	72.1	(-2.3)	84.26	(-3.35)	82.04	(-10.87)	11.78	(-2.08%)
DQN+I-SEE	**85.2**	**(+10.8, +13.1)**	**90.18**	**(+2.57, +5.92)**	92.59	(-0.32, **+10.55**)	**9.83**	**(+14.82%, +16.55%)**
GDPL	86.5		**94.97** best		83.90		7.64	
GDPL+Dvs.	72.8	(-13.7)	80.86	(-14.11)	81.10	(-2.80)	9.98	(-30.63)
GDPL+I-SEE	**93.2** best	**(+6.7, +20.4)**	91.83	(-3.14, **+10.97**)	**92.76**	**(+8.86, +11.66)**	**7.32** best	**(+4.19%, +26.65%)**

Table 3: Dialogue Effectiveness on Multiwoz. X+Dvs shows the improvement w.r.t. a baseline X. X+I-SEE reports the improvements w.r.t. X and X+Dvs, respectively.



(a) Learning curves of different ensemble size (E)

(b) Learning curves of different diversification horizons (H)

(c) Learning curves of different diversification ratios (η)

(d) Evaluating the Diversity of DUME

Figure 4: Experiment Results on Multiwoz.

rate. As we increase the size of the ensemble with $E = 1, 3, 5, 7, 9$, the degree of diversity increases. Figure 4a shows that initially increasing the diversity helps improve the performance; However, the trend turns downwards after reaching the optimum when $E = 5$. Figures 4b and 4c demonstrate similar trends. In the end, the best combined I-SEE setting is $E = 5, H = 5$, and $\eta = 0.2$. This experiment suggest that diversification can only help an agent's learning to a certain extent; Too much diversification beyond that may introduce too much noise in the learning and hurt the agent's performance. Therefore, the degree of diversification must be carefully chosen in practice.

5.4 Analysis of DUME

DUME is our collection of trainable diversified user models. We calculate the average pairwise KL-divergence for every two models M_{ϕ_i} and $M_{\phi_j} \in$ DUME to directly measure the degree of diversity within DUME. Each user model is run on the same stavte sequence $\{s^u_1, ..., s^u_t ...\}$ and outputs an action sequence $\{a'^u_1, ..., a'^u_t ...\}$. Since each a'^u_t may contain multiple dialogue acts, we break down every a'^u_t into individual dialogue acts and calculate the distribution over the dialogue act set A^u. The mean μ and standard deviation σ of the KL divergences are plotted in Figure 4d. We can see that as DUME has bigger size, both μ and σ increase; which means the differences between the DUME simulators dramatically increase and they would add much diversity into the agent's learning.

6 Conclusion

This paper presents Intermittent Short Extension Ensemble (I-SEE), a DRL diversification method that successfully improves dialogue diversity and policy robustness while maintaining high data quality. I-SEE uses an ensemble of trainable user models to achieve diversity and controls the diversification quality by branching from original dialogue trajectories only for a short horizon and intermittently. Our experiments on Multiwoz show that using I-SEE can significantly improve several best state-of-the-art DRL dialogue agents.

Acknowledgment

This research was supported by the U.S. National Science Foundation Grant no. IIS-145374. Any opinions, findings, conclusions, or recommendations expressed in this paper are of the authors, and do not necessarily reflect those of the sponsor.

References

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 475–484, New York, NY, USA. Association for Computing Machinery.

Marcin Andrychowicz, Dwight Crow, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. 2017. Hindsight experience replay. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5048–5058.

Ashutosh Baheti, Alan Ritter, Jiwei Li, and Bill Dolan. 2018. Generating more interesting responses in neural conversation models with distributional constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3970–3980, Brussels, Belgium. Association for Computational Linguistics.

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. 2018. Ask the right questions: Active question reformulation with reinforcement learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica S. Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 122–132. Association for Computational Linguistics.

Philipp Christmann, Rishiraj Saha Roy, Abdalghani Abujabal, Jyotsna Singh, and Gerhard Weikum. 2019. Look before you hop: Conversational question answering over knowledge graphs using judicious context expansion. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 729–738. ACM.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. 2018. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 4759–4770.

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. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. 2019. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12498–12509.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Çelikyilmaz. 2017. End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 733–743. Asian Federation of Natural Language Processing.

Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016b. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*.

Zachary C. 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 *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5237–5244. AAAI Press.

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

Jiyun Luo, Sicong Zhang, and Hui Yang. 2014. Win-win search: Dual-agent stochastic game in session search. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 587–596, New York, NY, USA. ACM.

Andrea Madotto, Samuel Cahyawijaya, Genta Indra Winata, Yan Xu, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Learning knowledge bases with parameters for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 2372–2394. Association for Computational Linguistics.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. 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 1468–1478. Association for Computational Linguistics.

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

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 845–854. Association for Computational Linguistics.

Tong Niu and Mohit Bansal. 2019. Automatically learning data augmentation policies for dialogue tasks. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 1317–1323. Association for Computational Linguistics.

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

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Nataniel Ruiz, Samuel Schulter, and Manmohan Chandraker. 2019. Learning to simulate. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Weiyan Shi, Kun Qian, Xuewei Wang, and Zhou Yu. 2019. How to build user simulators to train rl-based dialog systems. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 1990–2000. Association for Computational Linguistics.

Katherine Stasaski, Grace Hui Yang, and Marti A. Hearst. 2020. More diverse dialogue datasets via diversity-informed data collection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4958–4968, Online. Association for Computational Linguistics.

Shang-Yu Su, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Yun-Nung Chen. 2018. Discriminative deep dyna-q: Robust planning for dialogue policy learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3813–3823. Association for Computational Linguistics.

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press.

Ryuichi Takanobu, Runze Liang, and Minlie Huang. 2020. Multi-agent task-oriented dialog policy learning with role-aware reward decomposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 625–638. Association for Computational Linguistics.

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 2019, Hong Kong, China, November 3-7, 2019, pages 100–110. Association for Computational Linguistics.

Zhiwen Tang and Grace Hui Yang. 2020. Corpus-level end-to-end exploration for interactive systems. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2527–2534. AAAI Press.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 23–30. IEEE.

Faraz Torabi, Garrett Warnell, and Peter Stone. 2018. Behavioral cloning from observation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4950–4957. ijcai.org.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. *CoRR*, abs/1506.05869.

Zhenduo Wang and Qingyao Ai. 2021. Controlling the risk of conversational search via reinforcement learning. *CoRR*, abs/2101.06327.

Yen-Chen Wu, Bo-Hsiang Tseng, and Milica Gasic. 2020. Actor-double-critic: Incorporating model-based critic for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 854–863. Association for Computational Linguistics.

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4970–4977. AAAI Press.

Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul N. Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1933–1936. ACM.

Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. Convlab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 142–149. Association for Computational Linguistics.