MODELLING HIERARCHICAL STRUCTURE BETWEEN DIALOGUE POLICY AND NATURAL LANGUAGE GENERATOR WITH OPTION FRAMEWORK FOR TASK-ORIENTED DIALOGUE SYSTEM

Anonymous authors
Paper under double-blind review

ABSTRACT

Designing task-oriented dialogue systems is a challenging research topic, since it needs not only to generate utterances fulfilling user requests but also to guarantee the comprehensibility. Many previous works trained end-to-end (E2E) models with supervised learning (SL), however, the bias in annotated system utterances remains as a bottleneck. Reinforcement learning (RL) deals with the problem through using non-differentiable evaluation metrics (e.g., the success rate) as rewards. Nonetheless, existing works with RL showed that the comprehensibility of generated system utterances could be corrupted when improving the performance on fulfilling user requests. In o gur work, we (1) propose modelling the hierarchical structure between dialogue policy and natural language generator (NLG) with the option framework, called HDNO, where the latent dialogue act is applied to avoid designing specific dialogue act representations; (2) train HDNO via hierarchical reinforcement learning (HRL), as well as suggest the asynchronous updates between dialogue policy and NLG during training to theoretically guarantee their convergence to a local maximizer; and (3) propose using a discriminator modelled with language models as an additional reward to further improve the comprehensibility. We test HDNO on MultiWoz 2.0 and MultiWoz 2.1, the datasets on multi-domain dialogues, in comparison with word-level E2E model trained with RL, LaRL and HDSA, showing improvements on the performance evaluated by automatic evaluation metrics and human evaluation. Finally, we demonstrate the semantic meanings of latent dialogue acts to show the ability of explanation.

1 INTRODUCTION

Designing a task-oriented dialogue system is a popular and challenging research topic in the recent decades. In contrast to open-domain dialogue systems (Ritter et al., 2011), it aims to help people complete real-life tasks through dialogues without human service (e.g., booking tickets) (Young, 2006). In a task-oriented dialogue task, each dialogue is defined with a goal which includes user requests (i.e., represented as a set of key words known as slot values). The conventional dialogue system is comprised of 4 modules (see Appendix D.1), each of which used to be implemented with handcrafted rules (Chen et al., 2017). Given user utterances, it gives responses in turn to fulfill the requests via mentioning corresponding slot values.

Recently, several works focused on training a task-oriented dialogue system in end-to-end fashion (E2E) (Bordes et al., 2016; Wen et al., 2017) for generalizing dialogues outside corpora. To train a E2E model with supervised learning (SL), generated system utterances are forced to fit the oracle responses collected from human-to-human conversations (Budzianowski et al., 2017a). The oracle responses contain faults by humans thus being inaccurate, which leads to biased SL. On the other hand, the goal is absolutely clear, though the criterion of success rate that evaluates the goal completion is non-differentiable and cannot be used as a loss for SL.

To tackle this problem, reinforcement learning (RL) is applied to train a task-oriented dialogue system (Williams and Young, 2007; Zhao and Eskénazi, 2016; Peng et al., 2018; Zhao et al., 2019).

Specifically, some works merely optimized dialogue policy while other modules, e.g., the natural language generator (NLG), were fixed (Peng et al., 2018; Zhao et al., 2019; Su et al., 2018). In contrast, other works extended the dialogue policy to NLG and applied RL on the entire E2E dialogue system, regarding each generated word in a response as an action (Zhao and Eskénazi, 2016). Although previous works enhanced the performance on fulfilling user requests, the comprehensibility of generated system utterances are corrupted (Peng et al., 2018; Zhao et al., 2019; Tang et al., 2018a). The possible reasons are: (1) solely optimizing dialogue policy could easily cause the biased improvement on fulfilling user requests, ignoring the comprehensibility of generated utterances (see Appendix D.2); (2) the state space and action space (represented as a vocabulary) in E2E fashion is so huge that learning to generate comprehensible utterances becomes difficult (Lewis et al., 2017); and (3) dialogue system in E2E fashion may lack explanation during the procedure of decision.

In our work, we propose to model the hierarchical structure between dialogue policy and NLG with the option framework, i.e., a hierarchical reinforcement learning (HRL) framework (Sutton et al., 1999) called HDNO (see Section 4.1) so that the high-level temporal abstraction can provide the ability of explanation during the procedure of decision. Specifically, dialogue policy works as a high-level policy over dialogue acts (i.e. options) and NLG works as a low-level policy over generated words (i.e. primitive actions). Therefore, these two modules are decoupled during optimization with the smaller state space for NLG and the smaller action space for dialogue policy (see Appendix F). To reduce the efforts on designing dialogue act representations, we represent a dialogue act as latent factors. During training, we suggest the asynchronous updates between dialogue policy and NLG to theoretically guarantee their convergence to a local maximizer (see Section 4.2). Finally, we propose using a discriminator modelled with language models (Yang et al., 2018) as a reward to further improve the comprehensibility (see Section 5).

We evaluate HDNO on two datasets with dialogues in multiple domains: MultiWOZ 2.0 (Budzianowski et al., 2018) and MultiWOZ 2.1 (Eric et al., 2019), compared with word-level E2E (Budzianowski et al., 2018) trained with RL, LaRL (Zhao et al., 2019) and HDSA (Chen et al., 2019). The experiments show that HDNO works best on the total performance evaluated with automatic metrics (see Section 6.2.1) and the human evaluation (see Section 6.2.2). Finally, we study the latent dialogue acts and show that HDNO possesses the ability of explanation (see Section 6.2.3).

2 Related Work

Firstly, we go through the previous works on studying dialogue act representation for task-oriented dialogue system. Some previous works optimized dialogue policy with reinforcement learning (RL), which made decision via selecting from handcrafted dialogue acts represented as ontology (Peng et al., 2018; Young et al., 2007; Walker, 2000; He et al., 2018). Such a representation method is easily understood by human beings, while the dialogue act space becomes limited in representation. To deal with this problem, some researchers investigated training dialogue acts via fitting oracle dialogue acts represented in sequence (Chen et al., 2019; Zhang et al., 2019; Lei et al., 2018). This representation method generalized dialogue acts, however, designing a good representation is effort demanding. To handle this problem, learning a latent representation of dialogue act was attempted (Zhao et al., 2019; Yarats and Lewis, 2018). In our work, referring to (Zhao et al., 2019) we learn latent dialogue acts without any labels of dialogue act. By this view, our work can be regarded as an extension of LaRL (Zhao et al., 2019) on learning strategy.

Then, we review the previous works modelling a dialogue system with a hierarchical structure. In the field of task-oriented dialogue system, many works lay on modelling dialogue acts or the state space with a hierarchical structure to tackle the decision problem for dialogues with multi-domain tasks (Cuayáhuitl et al., 2009; Peng et al., 2017; Chen et al., 2019; Tang et al., 2018b; Budzianowski et al., 2017b). Distinguished from these works, our work views the relationship between dialogue policy and natural language generator (NLG) as a natural hierarchical structure and models it with the option framework (Sutton et al., 1999). In the field of open-domain dialogue system, a similar hierarchical structure was proposed (Serban et al., 2017; Saleh et al., 2019) but with a different motivation from ours. In this sense, these two fields are possible to be unified.

Finally, among the works training with hierarchical reinforcement learning (HRL), some of them set up an extrinsic reward for high-level policy and an intrinsic reward for low-level policy respectively to encourage the convergence (Peng et al., 2017; Budzianowski et al., 2017b). In our work, we

train both high-level policy and low-level policy with identical rewards to guarantee the consistency between two policies (Sutton et al., 1999).

3 BACKGROUND

Hierarchical Reinforcement Learning with Option Framework: Hierarchical reinforcement learning (HRL) is a variant of reinforcement learning (RL) which extends the decision problem to coarser grains with multiple hierarchies (Sutton et al., 1999; Dayan and Hinton, 1993; Parr and Russell, 1998; Dietterich, 1998). Amongst several HRL methods, the option framework (Sutton et al., 1999) is a temporal abstraction for RL, where each option (i.e. a high-level action) lasts for a number of steps through primitive actions (i.e. low-level actions). From the view of decision problems, a MDP defined with a fixed set of options naturally forms a semi-MDP (SMDP). Formally, an option $\boldsymbol{o} = \langle \mathbb{I}, \beta, \pi \rangle$ is composed of three components: an initiation set $\mathbb{I} \subseteq \mathbb{S}$ (where $\mathbb{S}$ is a state space), a termination function $\beta(\boldsymbol{s}_t) \mapsto [0, 1]$ and an intra-option policy $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t) \mapsto [0, 1]$. The reward over primitive actions is defined as $r_t \in \mathbb{R}$, identical to vanilla RL. An option $\boldsymbol{o}_t$ is available at $\boldsymbol{s}_t \in \mathbb{I}$. At each $\boldsymbol{s}_t$, π is used to decide a low-level action $\boldsymbol{a}_t$ until the option is stochastically terminated by β. Similar to RL on flat actions, the probability transition function over options is defined as $p(\boldsymbol{s}'|\boldsymbol{s}_t, \boldsymbol{o}_t) = \sum_{k=1}^{\infty} p(\boldsymbol{s}', k)\, \gamma^k$, where $p(s', k)$ is the probability of an option terminating in k steps and $\gamma \in (0, 1)$. The policy over options is defined as $\mu(\boldsymbol{o}_t|\boldsymbol{s}_t) \mapsto [0, 1]$ and the reward over options lasting for k steps is defined as $g(\boldsymbol{s}_t, \boldsymbol{o}_t, \boldsymbol{s}_{t+k}) = \mathbb{E}_\pi[\, \sum_{i=t}^{t+k} \gamma^{i-t}\, r_i \,]$ (abbreviated as g_t for simplicity). Given a set of options $o \in \mathbb{O}$, the optimization problem over options is defined as $\max_o \mathbb{E}_o[\, \sum_{k \in M} \gamma^{k-t} g_k \,]$, where $\boldsymbol{m} = (t, t', ...)$ is a sequence containing the time step of each event [1] that will be experienced from some time step t to future. To automatically fit complicated circumstances, we may also need to discover options dynamically during learning. Intra-option policy gradient theorem and termination policy gradient theorem (Bacon et al., 2017) provide the basis to apply a policy gradient method for option discovery.

4 MODELLING HIERARCHICAL STRUCTURE BETWEEN DIALOGUE POLICY AND NLG WITH OPTION FRAMEWORK

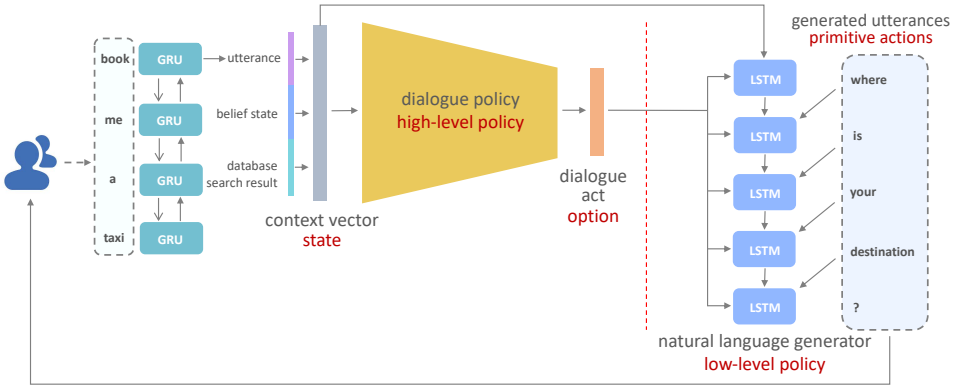


Figure 1: This diagram demonstrates the overall architecture for modelling the hierarchical structure between dialogue policy (i.e. high-level policy) and NLG (i.e. low-level policy) as an option framework. The text in gray represents the concepts for a traditional task-oriented dialogue system whereas the text in red matches the concepts for an option framework.

4.1 MODEL FORMULATION

In this section, we present a view on modelling the **H**ierarchical structure between **D**ialogue policy and **N**atural language generator (NLG) with the **O**ption framework (Sutton et al., 1999), called HDNO. Specifically, a dialogue act in HDNO is seen as an option whereas each generated word

[1] An event is defined as a policy over option calling an intra-option policy at some state.

from NLG becomes a primitive action. Accordingly, dialogue policy and NLG become the policy over option (i.e. high-level policy) and the intra-option policy (i.e. low-level policy) respectively. Distinguished from a conventional modular system, we additionally give a context to NLG to satisfy the option framework. Moreover, since the primitive action space (i.e. a vocabulary) comprises a termination symbol, NLG can take over the responsibility of termination. For this reason, termination policy is absorbed in the intra-option policy. HDNO is formally defined in Definition 1.

Definition 1. *A dialogue policy (i.e. a policy over option) is defined as $\phi : \mathbb{S} \times \mathbb{O} \rightarrow [0, 1]$, where $\mathbb{S}$ is a set of contexts (i.e. utterances, dialogue states and database search results); $\mathbb{O}$ is a set of dialogue acts (i.e. options). A dialogue act is defined as $\boldsymbol{o} = \langle \mathbb{I}_{\boldsymbol{o}}, \pi_{\boldsymbol{o}} \rangle$, where $\mathbb{I}_{\boldsymbol{o}} \subseteq \mathbb{S}$ is a set of corresponding contexts for a generated word and $\pi_{\boldsymbol{o}} : \mathbb{I}_{\boldsymbol{o}} \times \mathbb{V} \rightarrow [0, 1]$ is natural language generator (NLG) (i.e. an intra-option policy). $\mathbb{V}$ is a vocabulary (including a termination symbol).*

According to MDP theorem over option (Sutton et al., 1999) and intra-option policy gradient theorem (Bacon et al., 2017), we can naturally apply REINFORCE (Williams, 1992) to learn both ϕ and π. Therefore, following Section 3 and Definition 1, we can write policy gradients in our case such that $\nabla J(\phi) = \mathbb{E}_{\phi}[\ \sum_{k \in M} \gamma^{k-t} g_k \nabla \ln \phi(\boldsymbol{o}_t|\boldsymbol{s}_t)\ ]$ and $\nabla J(\pi_{\boldsymbol{o}_t}) = \mathbb{E}_{\pi_{\boldsymbol{o}_t}}[\ \sum_{i=t}^{T} \gamma^{i-t} r_i \nabla \ln \pi_{\boldsymbol{o}_t}(\boldsymbol{w}_t|\boldsymbol{s}_t)\ ]$, where we assume that the length of all generated system utterances is T and $\boldsymbol{m} = (t, t', ...)$ is a sequence containing the time steps of events that appear in future for an arbitrary $\boldsymbol{o}_t = \langle \mathbb{I}_{\boldsymbol{o}_t}, \pi_{\boldsymbol{o}_t} \rangle \in \mathbb{O}$.

4.2 Asynchronously Updating Dialogue Policy and NLG during Learning

As mentioned in Section 4.1, dialogue policy and NLG are represented by $\phi(\boldsymbol{o}|\boldsymbol{s})$ and $\pi_{\boldsymbol{o}}(\boldsymbol{w}|\boldsymbol{s})$ respectively. However, since $\boldsymbol{o} = \langle \mathbb{I}_{\boldsymbol{o}}, \pi_{\boldsymbol{o}} \rangle$, we can assume that when dialogue policy made a decision, it has to consider the current performance on the overall set of low-level policies for NLG, denoted as $\pi = \{\pi_{\boldsymbol{o}}\}_{\boldsymbol{o} \in \mathbb{O}}$. For the reason, we temporarily rewrite dialogue policy to $\phi(\boldsymbol{o}|\pi, \boldsymbol{s})$ for convenience. The aim is finding the best policies (i.e. maximizers) so that the value can be maximized such that $\max_{\phi,\pi} v(\boldsymbol{s}|\phi(\boldsymbol{o}|\pi, \boldsymbol{s})), \forall \boldsymbol{s} \in \mathbb{S}$. If updating these two policies synchronously, it will cause the composite state (i.e. $\langle \pi, \boldsymbol{s} \rangle$) of $\phi(\boldsymbol{o}|\pi, \boldsymbol{s})$ inconsistent before and after update each time. Therefore, the value does not always monotonically improve during learning, which will affect the convergence of both policies (see Proposition 1). To address this problem, we suggest updating dialogue policy and NLG asynchronously during learning to theoretically guarantee the convergence of these policies to a local maximizer (see Proposition 2).

Proposition 1. *Following the model of Definition 1, if $\phi(\boldsymbol{o}|\pi, \boldsymbol{s})$ and π are synchronously updated, the value does not always monotonically improve and the policies may never converge to a local maximizer.*

Proof. The complete proof is shown in Appendix A.1. □

Assumption 1. *(1) Reward function is bounded. (2) With sufficient number of samples, the Monte Carlo estimation for value on any state is accurate enough.*

Proposition 2. *Following the model of Definition 1 and Assumption 1, if $\phi(\boldsymbol{o}|\pi, \boldsymbol{s})$ and π are asynchronously updated, the value can improve monotonically during learning and the policies can finally converge to a local maximizer.*

Proof. The complete proof is shown in Appendix A.2. □

4.3 Implementation of HDNO

In implementation of HDNO, we represent a dialogue act as latent factors (Zhao et al., 2019), which reduces the effort on designing a suitable representation. In details, a dialogue act $\boldsymbol{z}$ (i.e. an indicator representing an option) is sampled from a dialogue policy represented as an isotropic multivariate Gaussian distribution such that $\phi(\boldsymbol{z}|\boldsymbol{c}; \lambda) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}(c), \boldsymbol{\Sigma}(\boldsymbol{c}))$, where $\boldsymbol{c}$ is a context as well as $\boldsymbol{\mu}(\boldsymbol{c}) \in \mathbb{R}^K$ and $\boldsymbol{\Sigma}(\boldsymbol{c}) \in \mathbb{R}^{K \times K}$ are parameterized with λ. Furthermore, NLG, i.e. $\pi(\boldsymbol{w}_t|\boldsymbol{z}, \tilde{\boldsymbol{c}}_t; \nu)$, is represented as a categorical distribution over a word parameterized with ν, conditioned on an option $\boldsymbol{z}$ and a context $\tilde{\boldsymbol{c}}_t$ which involves preceding generated utterances in addition to the context $\boldsymbol{c}$ that activates the option $\boldsymbol{z}$. The full picture of this architecture is shown in Figure 1.

Furthermore, $\phi(\boldsymbol{z}|\boldsymbol{c};\lambda)$ (i.e. dialogue policy) is implemented with one-layer linear model and outputs the mean and variance of a multivariate Gaussian distribution. The input of $\phi(\boldsymbol{z}|\boldsymbol{c};\lambda)$ is a context vector. In details, the last user utterances are firstly encoded with a bidirectional RNN (Schuster and Paliwal, 1997) with gated recurrent unit (GRU) cell (Chung et al., 2014) and global type attention mechanism (Bahdanau et al., 2015). Then, an oracle dialogue state and an oracle database search result are concatenated to an encoding vector of user utterances to form a context vector. The utterance encoder is only trained during pretraining and fixed as a context extractor during HRL, so that the context space is reduced. On the other hand, $\pi(\boldsymbol{w}_t|\boldsymbol{z},\tilde{\boldsymbol{c}}_t;\nu)$ (i.e. NLG) is implemented with a recurrent neural network (RNN) with long short-term memory (LSTM) cell (Hochreiter and Schmidhuber, 1997), where the initial state is the concatenation of a context vector and a dialogue act sampled from dialogue policy. The information in the initial state is assumed to be propagated to the hidden states at the future time steps, so we only feed it in the initial state in implementation.

4.4 Pretraining with Bayesian Framework

Compared to VHRED (Serban et al., 2017) proposed in the field of open-domain dialogue systems, the latent dialogue act in HDNO is equivalent to the latent variables in VHRED. However, a context of HDNO includes not only user utterances but also dialogue state and database search result. In this sense, HDNO extends VHRED to the field of task-oriented dialogue systems. As a result, by changing user utterances in VHRED to the context in HDNO, we can directly formulate a variational lower bound following the Bayesian framework and the model in Section 4.1 such that

$$\max_{\lambda,\nu} \mathbb{E}_{\boldsymbol{z}\sim\phi(\boldsymbol{z}|\boldsymbol{c};\lambda)}[\,\sum_{t=1}^{T}\log\pi(\boldsymbol{w}_t|\boldsymbol{z},\tilde{\boldsymbol{c}}_t;\nu)\,] - \beta\,\mathrm{KL}[\,\phi(\boldsymbol{z}|\boldsymbol{c};\lambda)\,||\,\mathcal{N}(\boldsymbol{z}|\boldsymbol{0},\boldsymbol{I})\,], \tag{1}$$

where $\phi(\boldsymbol{z}|\boldsymbol{c};\lambda)$ is constrained by KL-divergence (Kullback and Leibler, 1951) from a multivariate standard Gaussian distribution $\mathcal{N}(\boldsymbol{z}|\boldsymbol{0},\boldsymbol{I})$. Referring to (Higgins et al., 2017), we additionally add a multiplier β on the term of KL-divergence to control the disentanglement of a latent dialogue act $\boldsymbol{z}$. We use Eq.1 for pretraining dialogue policy and NLG in E2E fashion with oracle system utterances to roughly allocate the roles of these two modules. Nevertheless, restricted with the existing human faults in oracle system utterances (see Section 1), we need to further improve it via HRL.

5 Using Discriminator of Language Models as a Reward

Benefiting from the availability of non-differentiable evaluation metrics in RL, we can directly apply the success rate on reward denoted by r^{succ}. However, there exists two potential issues: (1) since the success rate is given until the end of a dialogue that is zero in the other turns, which may cause a sparse reward; and (2) the success rate is only correlated to the occurrence of request slots in generated system utterances, the improvement on comprehensibility may be weakened. To mitigate the above drawbacks, we propose to leverage a discriminator modelled as language models (see Definition 2) as an additional reward. Specifically, at each time step it evaluates each generated word by log-likelihood to reflect the comprehensibility.

Definition 2. *Discriminator $D(\boldsymbol{w}_t|\boldsymbol{w}_{t-1}) \mapsto [0,1]$ is defined as the Markov language model following (Yang et al., 2018). At some time step τ, for an arbitrary option $\boldsymbol{e} = \langle \mathbb{I}_e, \pi_e \rangle$, $\boldsymbol{w}_\tau \sim \pi_e(\cdot|\boldsymbol{s}_\tau)$ is a sampled word at state $\boldsymbol{s}_\tau \in \mathbb{I}_e$. The reward of discriminator to evaluate $\boldsymbol{w}_\tau$ is defined as $r_\tau^{disc} = \log D(\boldsymbol{w}_\tau|\boldsymbol{w}_{\tau-1})$, where $\boldsymbol{w}_{\tau-1}$ is a generated word at time step $\tau-1$.*

According to Definition 2, we can see that $\sum_{t=0}^{T'}\gamma^t r^{\text{disc}}$ consistently grows as the joint log-likelihood $\sum_{t=0}^{T'}\log D(\boldsymbol{w}_t|\boldsymbol{w}_{t-1})$ grows, thereby maximizing $\mathbb{E}_\pi[\,\sum_{t=0}^{T'}\gamma^t r_t^{\text{disc}}\,]$ is equivalent to maximizing $\mathbb{E}[\,\sum_{t=0}^{T'}\log D(\boldsymbol{w}_t|\boldsymbol{w}_{t-1})\,]$, where T' denotes the end of a turn. For this reason, r^{disc} is suitable for evaluating the comprehensibility of generated system utterances if we presume that the discriminator can well represent human language. Combining the reward of success rate r^{succ} and the reward of discriminator r^{disc}, we propose a total reward such that

$$r_t^{\text{total}} = (1-\alpha)\,r_t^{\text{succ}} + \alpha\,r_t^{\text{disc}}, \tag{2}$$

where $\alpha \in [0,1]$ is a multiplier controlling the trade-off between these two types of rewards.

6 EXPERIMENTS

6.1 EXPERIMENTAL SETUPS

Dataset Description: To evaluate the performance of our task-oriented dialogue system, we run experiments on the latest benchmark datasets MultiWoz 2.0 (Budzianowski et al., 2018) and MultiWoz 2.1 (Eric et al., 2019). MultiWoz 2.0 is a large scale task-oriented dialogue dataset including 10425 dialogues that spans 7 distinct domains, where the whole dialogues are generated by human-to-human conversations. Each dialogue is defined with a goal for a user, which may be consisting of 1-5 domains. A dialogue system attempts to fulfill a goal by interacting with a user. As for data preprocessing, we follow the same delexicalized method provided by (Budzianowski et al., 2018), also used in other works (Zhao et al., 2019; Chen et al., 2019). On the other hand, MultiWoz 2.1 is a modified version of MultiWoz 2.0 which mainly fixes the noisy dialogue state annotations and corrects 146 dialogue utterances. Finally, the dataset of either MultiWoz 2.0 or MultiWoz 2.1 is split into a training set with 8438 dialogues, a validation set with 1000 dialogues and a testing set with 1000 dialogues (Budzianowski et al., 2018; Eric et al., 2019).

Task Description: Since we only concentrate on learning dialogue policy and natural language generator (NLG), all experiments are conducted on the dialog-context-to-text generation task proposed in (Budzianowski et al., 2018). In this task, it assumes that a dialogue system has access to the oracle dialogue state and database search result. Given user utterances, a dialogue system attempts to generate appropriate utterances as a response in each turn. To train a dialogue system with hierarchical reinforcement learning (HRL), we follow the setups described in Definition 1. Each dialogue is only evaluated with the goal (e.g. calculating the success rate) at the end of dialogue, which means that no evaluation is allowed during any of turns.

Automatic Evaluation Metric: Following (Budzianowski et al., 2017a), we leverage three automatic metrics to evaluate generated utterances from a dialogue system such as inform rate, success rate and BLEU score. Inform rate measures whether a dialogue system provides appropriate entities (e.g., the name of restaurant). Success rate shows the ratio of request slots appearing in generated utterances. BLEU score (Papineni et al., 2002) evaluates the comprehensibility of generated utterances. Finally, we use a popular total score (Zhang et al., 2019; Mehri et al., 2019) such that $0.5 \times (\text{Inform} + \text{Success}) + \text{BLEU}$ to fairly evaluate the performance of a dialogue system.

Baseline Description: We compare HDNO with other models such as LaRL (Zhao et al., 2019), HDSA (Chen et al., 2019), and a baseline end-to-end model (Budzianowski et al., 2018). All these models leveraged oracle dialogue states and database search results, which are introduced as follows:

- The baseline end-to-end model (Budzianowski et al., 2018) is a model which directly maps a context to system utterances. Followed by (Zhao et al., 2019), we train it with RL, where each generated word is looked as an action. For convenience, we name it as word-level E2E model, abbreviated as WE2E.
- LaRL (Zhao et al., 2019) is a model that firstly represents a dialogue act as latent factors in the field of task-oriented dialogue system. Specifically, it models a latent dialogue act as categorical variables, each of which is mapped to a continuous embedding vector for learning. During training, it only updates dialogue policy, where a latent categorical dialogue act for each turn is looked as an action.
- HDSA (Chen et al., 2019) is a model that represents each dialogue act as a hierarchical graph. To fit the oracle dialogue act, a pretrained 12-layer BERT (Devlin et al., 2019) is applied. Then the predicted dialogue act is transformed to the hierarchical graph structure with 3-layer self-attention model (Vaswani et al., 2017), called disentangled self-attention model. This model is trained only with supervised learning (SL).

Experimental Details: For HDNO, we pretrain a model following Eq.1 and select the best model with the minimum loss on the validation set; as well as the discriminator is pretrained with oracle system utterances. During HRL, we initialize parameters with the pretrained model and select the best model according to the greatest reward on the validation set. For efficiency, we only use greedy search for decoding in validation. In test, we apply beam search (Medress et al., 1977) for decoding to obtain a better performance. The beam width is selected through the best validation performance for each model. For simplicity, we only show the performance of the best beam width on test set.

We use stochastic gradient descent (SGD) for HRL and Adam optimizer (Kingma and Ba, 2015) for pretraining. For the baselines, we train them with the original source codes. The specific details and hyperparameters for training and testing are shown in Appendix B.3.

Notification: Please notice that the results of baselines showed in their original papers could be underestimated, which is due to the upgrade of the official evaluator this year [2]. For this reason, we re-run these experiments with the original open source codes and evaluate the performance of all models (including HDNO) via the latest official evaluator in this work.

6.2 Results

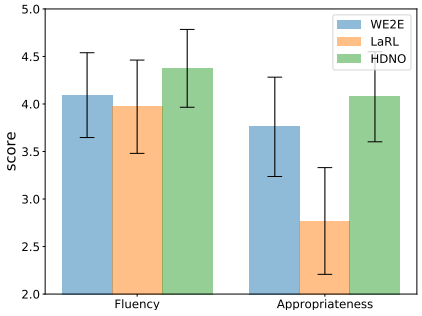
6.2.1 Comparison with Baselines

Table 1: The table shows the main results on MultiWoz 2.0 and MultiWoz 2.1.

	MultiWoz 2.0				MultiWoz 2.1			
	Inform(%)	**Success(%)**	**BLEU(%)**	**Total**	**Inform(%)**	**Success(%)**	**BLEU(%)**	**Total**
Human	91.00	82.70	-	-	86.30	79.10	-	-
WE2E	90.29	**86.59**	14.08	102.52	90.89	83.58	14.52	101.76
LaRL	93.49	84.98	12.01	101.25	92.39	**85.29**	13.72	102.56
HDSA	88.90	73.40	**23.15**	104.30	85.60	75.50	**21.57**	102.12
HDNO	**96.40**	84.70	18.85	**109.40**	**92.80**	83.00	18.97	**106.77**

We firstly compare HDNO with the state-of-the-art baselines and the human performance on both datasets. As Table 1 shows, we can see that HDNO (i.e., with $\alpha = 0.0001$ for MultiWoz 2.0 and $\alpha = 0.01$ for MultiWoz 2.1) largely outperforms the baselines on the inform rate and total score, while its BLEU score is lower than that of HDSA. Moreover, the performance of all models trained with RL on the inform rate and success rate exceeds that of the model trained with SL (i.e., HDSA). The possible reason may be that SL is highly dependent on the oracle system utterances and humans may commit faults during generating these dialogues as we stated in Section 1. Besides, this could explain the reason why all RL methods outperform the human performance. We also conducted experiments on comparing the proposed reward shape in Eq.2 of different α with other reward shapes (i.e., success rate + BLEU and success rate). The result shows that the proposed reward shape actually improves the performance (see Appendix B.2).

6.2.2 Human Evaluation



(a) Statistical results.

Example 1	
Context	[user]: I need to reserve a table at Steak House, can you help me ?
	[sys]: I sure can . Steak Houseis an expensive English restaurant in the center of town . What day would you like your reservation ?
	[user]: No hold off on booking for now . Can you help me find an attraction called Tate Modern?
Generated Response	WE2E: I can book that for you . What day and time would you like to book the table for ?
	LaRL: Your reference number is R01 .
	HDNO: What time would you like to dine ?
Example 2	
Context	[user]: Before booking , i would also like to know the travel time , price , and departure time please .
	[sys]: There are two trains meeting your needs with the first leaving at 16:00 and the last one leaving at 16:30. Do you want to book one of these ?
	[user]: No hold off on booking for now . Can you help me find an attraction called Tate Modern?
Generated Response	WE2E: Night Pubis a nightclub located in the center of town . The address is Street 1st. The phone number is 12345678. The entrance fee is 30pounds . The postcode is P01. Is there anything else i can help you with ?
	LaRL:Tate Modern s a museum in the center. Their phone number is 12345678.
	HDNO: Tate Modern s a museum in the center. Their phone number is 12345678. Is there anything else i can help you with ?

(b) Example generated responses.

Figure 2: (a) This figure shows the statistical results of human evaluation, where 30 people participated in this human evaluation and the questionnaire is constituted of 31 randomly selected turns of dialogues. (b) This figure shows the example generated responses from the questionnaire.

Due to the possible inconsistency between automatic evaluation metrics and human perception, we conducted human evaluation on comparing the quality of generated responses. We provide

[2]Please check it via the clarification for this incident below the table called Policy Optimization on the official website of MultiWoz: `https://github.com/budzianowski/multiwoz`.

two criteria for human assessors to evaluate the generated responses: (1) fluency: how fluent the generated responses are (i.e., with no obvious grammar errors and redundant descriptions); and (2) appropriateness: how related the generated responses are to the provided context. For each criterion, a human assessor is allowed to give a score ranging from 1 to 5, where 5 indicates the best and 1 indicates the worst performance. Then, we calculated the mean and the variance for the score of each model. The final results and some example generated responses from the questionnaire are shown in Figure 2. The proposed HDNO performs best in the human evaluation, especially outperforming LaRL (Zhao et al., 2019) on appropriateness so much.

6.2.3 Study on Latent Dialogue Act

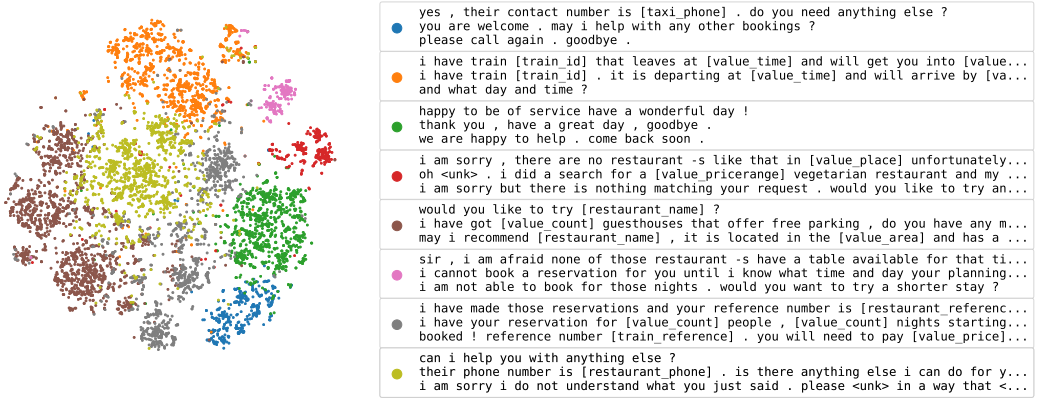


Figure 3: This diagram demonstrates latent dialogue acts clustered in 8 categories on a 2-D plane. Clustering is conducted with k-means algorithm (Arthur and Vassilvitskii, 2006) on the original dimensions whereas dimension reduction is conducted with T-SNE algorithm (Maaten and Hinton, 2008). We randomly select 3 turns of system utterances for each cluster.

We now study the semantic meanings of latent dialogue acts and demonstrate the clustering results as Figure 3 shows. Through analyzing the randomly selected system utterances, we find some semantic meanings for each cluster of latent dialogue acts. Specifically, the clusters in blue dots and green dots are related to the general phrases for goodbye at the end of service; the cluster in orange dots is related to the booking for trains; the cluster in red dots is related to informing user with database search results; the cluster in brown dots is related to recommendation; the cluster in pink dots is related to informing unsuccessful booking; the cluster in grey is related to informing booked; and the cluster in yellow is related to requesting more information. Surprisingly, these semantic meanings of latent dialogue acts are highly correlated with that of the oracle handcrafted dialogue acts (see Appendix E) described by Budzianowski et al. (2018). Therefore, learning latent dialogue act with the option framework (Sutton et al., 1999) may potentially substitute for handcrafting dialogue act with ontology, without losing its explanation.

7 Conclusion and Future Work

In this paper, we present a view on modelling the hierarchical structure between dialogue policy and natural language generator (NLG) with the option framework (Sutton et al., 1999) and train it with hierarchical reinforcement learning (HRL). Moreover, we suggest asynchronous updates between dialogue policy and NLG to theoretically guarantee their convergence to a local maximizer. Finally, we propose using a discriminator modelled as language models (Yang et al., 2018) as a reward to further improve the comprehensibility of generated responses.

In the future work, we are going to extend this work to optimizing all modules instead of only dialogue policy and NLG by RL, as well as study on solving out the problem on credit assignment among these modules (Chen et al., 2017) during training. Moreover, thanks to the option framework (Sutton et al., 1999), the latent dialogue act shows explicit semantic meanings, while disentangling factors of a latent dialogue act (i.e. each latent factor owning a distinct semantic meaning) during HRL is left to be further investigated.

REFERENCES

Alan Ritter, Colin Cherry, and William B Dolan. Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing*, pages 583–593. Association for Computational Linguistics, 2011.

Steve Young. Using pomdps for dialog management. In *2006 IEEE Spoken Language Technology Workshop*, pages 8–13. IEEE, 2006.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35, 2017.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*, 2016.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina Maria Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve J. Young. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 438–449. Association for Computational Linguistics, 2017.

Paweł Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Inigo Casanueva, Lina Rojas-Barahona, and Milica Gašić. Sub-domain modelling for dialogue management with hierarchical reinforcement learning. *arXiv preprint arXiv:1706.06210*, 2017a.

Jason D Williams and Steve Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007.

Tiancheng Zhao and Maxine Eskénazi. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, pages 1–10. The Association for Computer Linguistics, 2016.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. 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, 2018.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. *arXiv preprint arXiv:1902.08858*, 2019.

Shang-Yu Su, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Yun-Nung Chen. 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, 2018.

Da Tang, Xiujun Li, Jianfeng Gao, Chong Wang, Lihong Li, and Tony Jebara. Subgoal discovery for hierarchical dialogue policy learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2298–2309. Association for Computational Linguistics, 2018a.

Mike Lewis, Denis Yarats, Yann N. Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2443–2453. Association for Computational Linguistics, 2017.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, pages 7287–7298, 2018.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5016–5026. Association for Computational Linguistics, 2018.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*, 2019.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. 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 3696–3709. Association for Computational Linguistics, 2019.

Steve Young, Jost Schatzmann, Karl Weilhammer, and Hui Ye. The hidden information state approach to dialog management. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–149. IEEE, 2007.

Marilyn A Walker. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Artificial Intelligence Research*, 12:387–416, 2000.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2333–2343. Association for Computational Linguistics, 2018.

Yichi Zhang, Zhijian Ou, and Zhou Yu. Task-oriented dialog systems that consider multiple appropriate responses under the same context. *arXiv preprint arXiv:1911.10484*, 2019.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, 2018.

Denis Yarats and Mike Lewis. Hierarchical text generation and planning for strategic dialogue. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5587–5595. PMLR, 2018.

Heriberto Cuayáhuitl et al. *Hierarchical reinforcement learning for spoken dialogue systems*. PhD thesis, The University of Edinburgh, 2009.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Çelikyilmaz, Sungjin Lee, and Kam-Fai Wong. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2231–2240. Association for Computational Linguistics, 2017.

Da Tang, Xiujun Li, Jianfeng Gao, Chong Wang, Lihong Li, and Tony Jebara. Subgoal discovery for hierarchical dialogue policy learning. *arXiv preprint arXiv:1804.07855*, 2018b.

Pawel Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrksic, Tsung-Hsien Wen, Iñigo Casanueva, Lina Maria Rojas-Barahona, and Milica Gasic. Sub-domain modelling for dialogue management with hierarchical reinforcement learning. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, pages 86–92. Association for Computational Linguistics, 2017b.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Abdelrhman Saleh, Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, and Rosalind Picard. Hierarchical reinforcement learning for open-domain dialog. *arXiv preprint arXiv:1909.07547*, 2019.

Peter Dayan and Geoffrey E Hinton. Feudal reinforcement learning. In *Advances in neural information processing systems*, pages 271–278, 1993.

Ronald Parr and Stuart J Russell. Reinforcement learning with hierarchies of machines. In *Advances in neural information processing systems*, pages 1043–1049, 1998.

Thomas G Dietterich. The maxq method for hierarchical reinforcement learning. In *ICML*, volume 98, pages 118–126. Citeseer, 1998.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *Iclr*, 2(5):6, 2017.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. Structured fusion networks for dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 165–177, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Mark F. Medress, Franklin S Cooper, Jim W. Forgie, CC Green, Dennis H. Klatt, Michael H. O'Malley, Edward P Neuburg, Allen Newell, DR Reddy, B Ritea, et al. Speech understanding systems: Report of a steering committee. *Artificial Intelligence*, 9(3):307–316, 1977.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Heinz H Bauschke, Patrick L Combettes, et al. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

Jiahuan Pei, Pengjie Ren, and Maarten de Rijke. A modular task-oriented dialogue system using a neural mixture-of-experts. *arXiv preprint arXiv:1907.05346*, 2019a.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*, 2020.

Jiahuan Pei, Pengjie Ren, Christof Monz, and Maarten de Rijke. Retrospective and prospective mixture-of-generators for task-oriented dialogue response generation. *arXiv preprint arXiv:1911.08151*, 2019b.

Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. Alternating recurrent dialog model with large-scale pre-trained language models. *arXiv preprint arXiv:1910.03756*, 2019.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. Soloist: Few-shot task-oriented dialog with a single pre-trained auto-regressive model. *arXiv preprint arXiv:2005.05298*, 2020.

Kai Wang, Junfeng Tian, Rui Wang, Xiaojun Quan, and Jianxing Yu. Multi-domain dialogue acts and response co-generation. *arXiv preprint arXiv:2004.12363*, 2020.

S Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

APPENDICES

A PROOFS

A.1 PROOF OF PROPOSITION 1

Proposition 1. *Following the model of Definition 1, if $\phi(\boldsymbol{o}|\pi, \boldsymbol{s})$ and π are synchronously updated, the value does not always monotonically improve and the policies may never converge to a local maximizer.*

Proof. At an arbitrary time step $t \in \mathbb{N}$, assume that both of $\phi(\boldsymbol{o}|\pi, \boldsymbol{s})$ and π have been updated for $q \in \mathbb{N}$ times, denoted as $\phi^q(\pi^q)$ and π^q respectively for conciseness. The current value for any arbitrary state $\boldsymbol{s} \in \mathbb{S}$ is denoted as $v\big(\boldsymbol{s}|\phi^q(\pi^q)\big)$. If we synchronously update $\phi^q(\pi^q)$ and π^q, we can obtain the following equations on value after updates at time step $t \in \mathbb{N}$ such that

$$\begin{cases} v^t\big(\boldsymbol{s}|\phi^{q+1}(\pi^q)\big) \geq v^t\big(\boldsymbol{s}|\phi^q(\pi^q)\big), \\ v^t\big(\boldsymbol{s}|\phi^q(\pi^{q+1})\big) \geq v^t\big(\boldsymbol{s}|\phi^q(\pi^q)\big). \end{cases} \tag{3}$$

At next time step $t + 1$, however, the actual value for any arbitrary state $\boldsymbol{s}$ that we obtain from the last synchronous update is the equation such that

$$v^{t+1}\big(\boldsymbol{s}|\phi^{q+1}(\pi^{q+1})\big), \tag{4}$$

and the following scenario such that

$$v^{t+1}\big(\boldsymbol{s}|\phi^{q+1}(\pi^{q+1})\big) < v^{t+1}\big(\boldsymbol{s}|\phi^{q}(\pi^{q})\big) \tag{5}$$

could happen, which means that the monotonic improvement path of the value cannot always hold during learning. Since the value is an evaluation of policies, the policies could never converge to a local maximizer. □

A.2 Proof of Proposition 2

Lemma 1 (Bauschke et al. (2011))**.** *(1) Any increasing bounded sequence $(x_n)_{n\in\mathbb{N}}$ in $\mathbb{R}$ is Fejér Monotone with respect to $\sup\{x_n\}_{n\in\mathbb{N}}$ such that $\forall n \in \mathbb{N},\ \ ||x_{n+1} - \sup\{x_n\}_{n\in\mathbb{N}}|| \leq ||x_n - \sup\{x_n\}_{n\in\mathbb{N}}||$. (2) For a Fejér Monotone Sequence $(x_n)_{n\in\mathbb{N}}$, $||x_n - \sup\{x_n\}_{n\in\mathbb{N}}||_{n\in\mathbb{N}}$ converges, $||\cdot||$ is an arbitrary norm in $\mathbb{R}$.*

Assumption 1. *(1) Reward function is bounded. (2) With sufficient number of samples, the Monte Carlo estimation for value on any state is accurate enough.*

Proposition 2. *Following the model of Definition 1 and Assumption 1, if $\phi(\boldsymbol{o}|\pi, \boldsymbol{s})$ and π are asynchronously updated, the value can improve monotonically during learning and both policies can finally converge to a local maximizer.*

Proof. Firstly, in REINFORCE (Williams, 1992) the value is approximated by Monte Carlo estimation. Following (2) in Assumption 1 and the denotation shown in the proof of Proposition 1 above, if we update $\phi(\pi)$ and π asynchronously every $k \in \mathbb{N}$ steps, from some time step $t \in \mathbb{N}$, assuming that both of policies have been updated $q \in \mathbb{N}$ times, we can construct a monotonically increasing sequence of values for any arbitrary state $\boldsymbol{s} \in \mathbb{S}$ during learning such that

$$\begin{aligned} &v^{t}\big(\boldsymbol{s}|\phi^{q}(\pi^{q})\big) \leq v^{t+k}\big(\boldsymbol{s}|\phi^{q+1}(\pi^{q})\big) \leq v^{t+2k}\big(\boldsymbol{s}|\phi^{q+1}(\pi^{q+1})\big) \\ &\leq v^{t}\big(\boldsymbol{s}|\phi^{q}(\pi^{q})\big) \leq ... \leq \begin{cases} v^{t+nk}\big(\boldsymbol{s}|\phi^{q+\frac{n}{2}}(\pi^{q+\frac{n}{2}})\big), & \text{if n is an even number,} \\ v^{t+nk}\big(\boldsymbol{s}|\phi^{q+\frac{n+1}{2}}(\pi^{q+\frac{n-1}{2}})\big), & \text{if n is an odd number.} \end{cases} \end{aligned} \tag{6}$$

Due to (1) in Assumption 1, we get that the value is bounded and we consider that $v(\boldsymbol{s}|\phi(\pi)) \in \mathbb{R}$, $\forall \boldsymbol{s} \in \mathbb{S}$. According to (1) Lemma 1, the sequence of values is Fejér Monotone with respect to the maximum value. For simplicity, we denote the sequence of values in Eq.6 as $\{v^m\}_{m\in\{t+nk|t,b,k\in\mathbb{N}\}}$ and the maximum value as v^*. Since (2) in Lemma 1, we can conclude that $\{||v^m - v^*||\}_{m\in\{t+nk|t,b,k\in\mathbb{N}\}}$ can converge. Also, since the v^* has to be the final item of the sequence, we can write that

$$\big|\big|\ ||v^m - v^*|| - ||v^* - v^*||\ \big|\big| \to 0,\ m \to \infty. \tag{7}$$

If we rearrange the left hand side of Eq.7, we can obtain the result such that

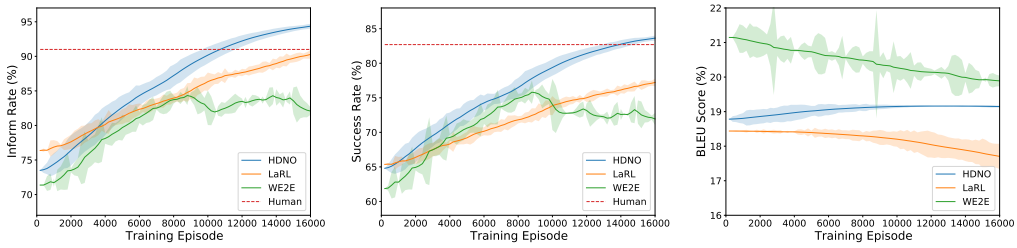
$$v^m \to v^*,\ m \to \infty. \tag{8}$$

From Eq.8, we can conclude that finally the sequence of $v(\boldsymbol{s}|\phi(\pi))$ will converge to some local maximum. Since the value is an evaluation of $\phi(o|\pi, \boldsymbol{s})$ and $\pi(\boldsymbol{w}|\boldsymbol{s})$, we can get the conclusion that the asynchronous updates enable these two policies to converge to a local maximizer. □

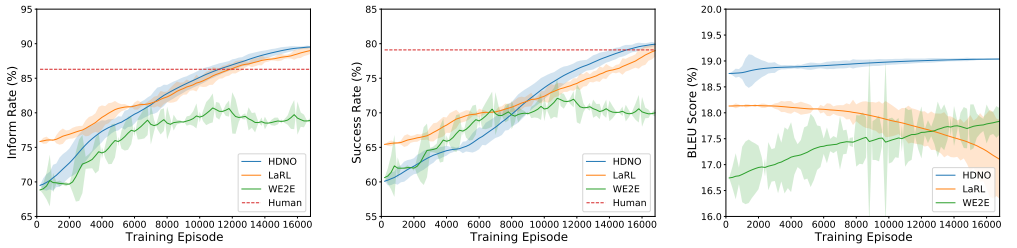
B Extra Experimental Results

B.1 Learning Curves

In this section, we show the learning curves of our model (i.e. HDNO) and baselines on both MultiWoz 2.0 and MultiWoz 2.1. To show the performance of generalization, we only demonstrate the validation results during training. As seen from Figure 4, compared with baselines our model can preserve the comprehensibility (i.e. BLEU) while improving success rate and inform rate faster. To make the figure more concise, we only show the results for the initial 16,000 episodes for conciseness of the figure.



(a) Inform rate during training on MultiWoz 2.0. (b) Success rate during training on MultiWoz 2.0. (c) BLEU score during training on MultiWoz 2.0.



(d) Inform rate during training on MultiWoz 2.1. (e) Success rate during training on MultiWoz 2.1. (f) BLEU score during training on MultiWoz 2.1.

Figure 4: Validation inform rate, success rate and BLEU score during training.

Table 2: The table shows the results of different reward shapes for HDNO on MultiWoz 2.0.

	α	**Inform(%)**	**Success(%)**	**BLEU(%)**	**Total**
Success	-	96.10	84.20	18.51	108.66
Success + BLEU	-	95.60	83.30	18.99	108.44
Success + Discriminator	0.0001	96.40	84.70	18.85	**109.40**
	0.0005	96.30	**84.90**	18.50	109.10
	0.001	96.20	84.20	**19.04**	109.24
	0.005	**97.00**	84.10	18.72	109.27

B.2 Study on Reward Shapes

We now compare the proposed reward shape in Eq.2 with a reward only constituted of the success rate and a reward combining the success rate with BLEU score (i.e. a linear combination like the proposed reward shape). As Table 2 shows, in comparison with other reward shapes, the proposed reward shape (i.e. success + discriminator) performs better on preserving the comprehensibility while improving the success rate and inform rate at maximum. On the other hand, the result of only using the success rate as reward still exceeds that of baselines shown in Table 1, which validates the analysis of alternating updates between dialogue policy and NLG stated in Section 4.2. To further study the impact of discriminator in the total reward, we also run several ablation studies on α (see Section 5). As Table 2 shows, the results oscillate within a small range if α is selected within a rational range.

B.3 Complete Results with Beam Search

In this section, we show the complete HRL results and the pretraining results with beam search in Table 3 and Table 4 respectively, where beam width is selected from 1, 2 and 5. Obviously, HRL gives an enormous improvement on the performance in comparison with the pretraining. Nevertheless, the pretraining is an essential part that cannot be replaced before training with reinforcement learning in task-oriented dialogue system. The possibility of training from scratch in HRL with a good performance for task-oriented dialogue system is left to be investigated.

Table 3: The table shows the HRL results with beam search on MultiWoz 2.0 and MultiWoz 2.1.

Reward Shapes	α	Beam width = 1				Beam width = 2				Beam width = 5			
		Inform(%)	Success(%)	BLEU(%)	Total	Inform(%)	Success(%)	BLEU(%)	Total	Inform(%)	Success(%)	BLEU(%)	Total
MultiWoz 2.0 (Success + Discriminator)	0.0001	96.70	84.50	18.72	109.32	96.40	84.70	18.85	109.40	96.30	84.30	18.70	109.00
	0.0005	96.30	83.60	18.60	108.55	96.40	84.10	18.30	108.61	96.30	84.90	18.50	109.10
	0.001	95.50	83.90	19.29	108.99	96.20	84.20	19.04	109.24	95.70	84.40	18.80	108.85
	0.005	97.00	84.10	18.72	109.27	96.50	84.30	18.78	109.18	96.80	83.90	18.70	109.05
MultiWoz 2.0 (Success)	-	95.70	82.00	18.83	107.68	96.10	84.20	18.51	108.66	95.30	82.70	18.57	107.57
MultiWoz 2.0 (Success + BLEU)	-	94.90	81.20	19.19	107.24	95.60	83.30	18.99	108.44	95.70	83.20	18.78	108.23
MultiWoz 2.1 (Success + Discriminator)	0.01	92.50	82.50	19.16	106.66	92.60	80.80	19.09	105.79	92.80	83.00	18.97	106.87

Table 4: The table shows the pretraining results with beam search on MultiWoz 2.0 and MultiWoz 2.1.

	MultiWoz 2.0				MultiWoz 2.1			
	Inform(%)	Success(%)	BLEU(%)	Total	Inform(%)	Success(%)	BLEU(%)	Total
Beam=1	69.50	62.00	19.10	84.85	70.70	61.40	18.24	84.29
Beam=2	67.90	60.20	19.00	83.05	71.50	62.20	19.28	86.13
Beam=5	69.50	62.50	19.20	85.20	71.40	62.80	19.12	86.22

B.4 Complete Benchmark Results on MultiWoz 2.0

Table 5: The table shows the full benchmark results on MultiWoz 2.0, compared with HDNO.

	Inform(%)	Success(%)	BLEU(%)	Total
HDNO (our model)	**96.40**	**84.70**	18.85	**109.4**
TokenMoE (Pei et al., 2019a)	75.30	59.70	16.81	84.31
Baseline (Budzianowski et al., 2018)	71.29	60.96	18.80	84.93
Structured Fusion (Mehri et al., 2019)	82.70	72.10	16.34	93.74
LaRL (Zhao et al., 2019)	82.80	79.20	12.80	93.80
SimpleTOD (Hosseini-Asl et al., 2020)	88.90	67.10	16.90	94.90
MoGNet (Pei et al., 2019b)	85.30	73.30	20.13	99.43
HDSA (Chen et al., 2019)	82.90	68.90	**23.60**	99.50
ARDM (Wu et al., 2019)	87.40	72.80	20.60	100.70
DAMD (Zhang et al., 2019)	89.20	77.90	18.60	102.15
SOLOIST (Peng et al., 2020)	89.60	79.30	18.30	102.75
MarCo (Wang et al., 2020)	92.30	78.60	20.02	105.47

In this section, we show the complete state-of-the-art results on MultiWoz 2.0 for the policy optimization task, from the official records [3]. All these results are collected from their original papers. As Table 5 shows, our model (i.e. HDNO) leads the board on the total performance evaluated by $0.5 \times (\text{Inform} + \text{Success}) + \text{BLEU}$, where each metric is measured in percentage.

B.5 Examples of Generated Delexicalized Dialogues

In this section, we demonstrate some system utterances generated by the baselines and our model (i.e. HDNO). Since most of the dialogues in MultiWoz 2.0 and MultiWoz 2.1 are similar, we only show the results on MultiWoz 2.0. As we can see from Table 6 and 7, compared with the baselines trained with SL (i.e. HDSA), the performance of HDNO on fulfilling a user's request is actually better, however, the generated utterances of HDSA could be more fluent and comprehensible. In comparison with other baselines trained with RL, the generated utterances of HDNO is apparently more fluent and comprehensible. Especially, WE2E tends to generate as many slots as possible so as to increase the success rate, while regardless of the comprehensibility of generated utterances. This is the common issue of most RL methods on task-oriented dialogue system as we stated in Section 3 in the main part of paper.

[3] https://github.com/budzianowski/multiwoz.

Table 6: The table shows some delexicalized system utterances generated by the baselines and our model on MultiWoz 2.0.

Models	Delexicalized Dialogue
HDNO (Our Model)	*usr*: i need train reservations from [value_place] to [value_place] *sys*: what day would you like to travel ? *usr*: i would like to leave on [value_day] and arrive by [value_time] *sys*: [train_id] leaves at [value_time] and arrives at [value_time] . would you like me to book that for you ? *usr*: before booking , i would also like to know the travel time , price , and departure time please . *sys*: the travel time is [value_count] minutes . the price is [value_price] pounds . is there anything else i can help you with ? *usr*: no hold off on booking for now . can you help me find an attraction called [attraction_name] ? *sys*: [attraction_name] is a museum in the [value_area] . would you like their location ? *usr*: yes , that was all i needed . thank you very much ! *sys*: you are welcome . have a nice day . goodbye .
WE2E	*usr*: i need train reservations from [value_place] to [value_place] *sys*: i have [value_count] trains leaving [value_place] on [value_day] . what day and time would you like to travel ? *usr*: i would like to leave on [value_day] and arrive by [value_time] *sys*: the [train_id] arrives at [value_time] . the price is [value_price] pounds . the train id is [train_id] . the price is [value_price] pounds . the train id is [train_id] . the train id is [train_id] . the train id is [train_id] ... *usr*: before booking , i would also like to know the travel time , price , and departure time please . *sys*: the train id is [train_id] and the price is [value_price] pounds . the train id is [train_id] . the price is [value_price] pounds . the train id is [train_id] . the train id is [train_id] . the train id is [train_id] ... *usr*: no hold off on booking for now . can you help me find an attraction called [attraction_name] ? *sys*: [attraction_name] is a nightclub located in the [value_area] of town . the address is [attraction_address] . the phone number is [attraction_phone] . the entrance fee is [value_count] pounds . the postcode is [attraction_postcode] ... *usr*: yes , that was all i needed . thank you very much ! *sys*: you are welcome . have a great day . goodbye .
LaRL	*usr*: i need train reservations from [value_place] to [value_place] *sys*: i have a train leaving at [value_time] . would you like me to book that for you ? *usr*: i would like to leave on [value_day] and arrive by [value_time] *sys*: i have a train leaving at [value_time] . would you like me to book that for you ? *usr*: before booking , i would also like to know the travel time , price , and departure time please . *sys*: the [train_id] leaves at [value_time] and arrives at [value_time] . the travel time is [value_count] minutes. *usr*: no hold off on booking for now . can you help me find an attraction called [attraction_name] ? *sys*: [attraction_name] is located in the [value_area] of town . the address is [attraction_address] and the postcode is [attraction_postcode] . *usr*: yes , that was all i needed . thank you very much ! *sys*: you are welcome . have a great day .
HDSA	*usr*: i need train reservations from [value_place] to [value_place] *sys*: what day and time would you like to travel ? *usr*: i would like to leave on [value_day] and arrive by [value_time] *sys*: what time would you like to leave ? *usr*: before booking , i would also like to know the travel time , price , and departure time please . *sys*: the train leaves at [train_leaveat] and arrives by [train_arriveby] . the price is [train_price] pounds and the travel time is [value_count] minutes . *usr*: no hold off on booking for now . can you help me find an attraction called [attraction_name] ? *sys*: it is located in the [attraction_area] of town at [attraction_address] . their phone number is [attraction_phone] . is there anything else i can help you with ? *usr*: yes , that was all i needed . thank you very much ! *sys*: you are welcome . have a great day !
Human	*usr*: i need train reservations from [value_place] to [value_place] *sys*: i have [value_count] trains matching your request . is there a specific day and time you would like to travel ? *usr*: i would like to leave on [value_day] and arrive by [value_time] *sys*: there are [value_count] trains for the day and time you request . would you like to book it now ? *usr*: before booking , i would also like to know the travel time , price , and departure time please . *sys*: there are [value_count] trains meeting your needs with the first leaving at [value_time] and the last [value_count] leaving at [value_time] . do you want to book [value_count] of these ? *usr*: no hold off on booking for now . can you help me find an attraction called [attraction_name] ? *sys*: yes it is a cinema located in the [value_area] part of town what information would you like on it ? *usr*: thank you for using our system . *sys*: you are welcome . have a nice day . goodbye .

C Extra Experimental Setups

C.1 Training Details

During pretraining, natural language generator (NLG) is trained with forcing the prediction at each time step to match an oracle word given a preceding oracle word as input. Nevertheless, during hierarchical reinforcement learning (HRL), NLG updates the estimated distribution at each time step by $\nabla J(\pi) = \mathbb{E}_{\pi}[\ \sum_{i=t}^{T} \gamma^{i-t} r_i \nabla \ln \pi(\boldsymbol{w}_t|\boldsymbol{s}_t)\ ]$ given a word sampled from a preceding predicted distribution. As for sampling a dialogue act from dialogue policy, we leverage reparameterization trick during pretraining. Finally, the discriminator is firstly pretrained and fixed as a reward during HRL.

We find that it is useful to train both of the discriminator and HDNO simultaneously during pretraining. Specifically, in addition to training on oracle system utterances, we also use the generated utterances to train the discriminator, which can improve its performance in experiments. The opti-

Table 7: The table shows some delexicalized system utterances generated by the baselines and our model on MultiWoz 2.0.

Models	**Delexicalized Dialogue**
HDNO (Our Model)	*usr*: please help me make a taxi reservation *sys*: what time would you like to leave ? *usr*: it does not matter , just so i arrive by [value_time] . *sys*: what is your destination ? *usr*: i want to go to the [restaurant_name] . and i need the car type and contact number please . *sys*: booking completed ! booked car type [taxi_type] contact number [taxi_phone] is there anything else i can help you with ? *usr*: no that is all . thanks *sys*: you are welcome . have a nice day . goodbye .
WE2E	*usr*: please help me make a taxi reservation *sys*: i can help with that . what time do you need to leave or arrive by ? *usr*: it does not matter , just so i arrive by [value_time] . *sys*: i would be happy to help with your request , but i will need to know what time you are looking for . *usr*: i want to go to the [restaurant_name] . and i need the car type and contact number please . *sys*: i have booked a [taxi_type] for you . the contact number is [taxi_phone] . is there anything else i can help you with today ? *usr*: no that is all . thanks *sys*: you are welcome . have a great day . goodbye .
LaRL	*usr*: please help me make a taxi reservation *sys*: i have booked a [taxi_type] for you . the contact number is [taxi_phone] . *usr*: it does not matter , just so i arrive by [value_time] . *sys*: i have booked a [taxi_type] for you . the contact number is [taxi_phone] . *usr*: i want to go to the [restaurant_name] . and i need the car type and contact number please . *sys*: i have booked a [taxi_type] for you . the contact number is [taxi_phone] . *usr*: no that is all . thanks *sys*: you are welcome . goodbye .
HDSA	*usr*: please help me make a taxi reservation *sys*: what time would you like to leave ? *usr*: it does not matter , just so i arrive by [value_time] . *sys*: where will you be departing from and what is your destination ? *usr*: i want to go to the [restaurant_name] . and i need the car type and contact number please . *sys*: your taxi is booked . the car type is a [taxi_type] and the contact number is [taxi_phone] . is there anything else i can help you with ? *usr*: no that is all . thanks *sys*: you are welcome . have a great day !
Human	*usr*: please help me make a taxi reservation *sys*: i can help you with that . when do you need to leave ? *usr*: it does not matter , just so i arrive by [value_time] . *sys*: i can help you with that . where are you going ? *usr*: i want to go to the [restaurant_name] . and i need the car type and contact number please . *sys*: okay , i have a [taxi_type] for you with the contact number [taxi_phone] . is there anything else you need today ? *usr*: no that is all . thanks *sys*: you are welcome , have a wonderful time !

mization problem is expressed as Eq.9, where $D(\cdot|\cdot,\theta)$ is the discriminator defined in Definition 2 in the main part of paper, parameterized with θ; and η is a multiplier to control the contribution of the generated utterances $(\hat{\boldsymbol{w}}_t)_{t=0}^{T'}$ given an initial state. We hypothesize that the generated utterances could expand the original small scale corpus (i.e. oracle system utterances), so as to improve the generalization of the discriminator. Since we have not grasped the intrinsic reason of this method, it is only regarded as a trick for training the discriminator in our work.

$$\max_{\theta} \sum_{t=0}^{T} \log D(\boldsymbol{w}_t|\boldsymbol{w}_{t-1},\theta) + \eta \sum_{t=0}^{T'} \log D(\hat{\boldsymbol{w}}_t|\hat{\boldsymbol{w}}_{t-1},\theta) \tag{9}$$

During HRL, r^{succ} and r^{disc} are respectively normalized by z-score normalization (Patro and Sahu, 2015) to adaptively control their impacts in the total reward. For instance, when either r^{succ} or r^{disc} converges around some value, its normalized value will be close to zero and the total reward will be biased to the other. Therefore, this can further mitigate the conflicts between improving both the success rate and preserving the comprehensibility as stated in Section 3 in the main part of paper.

Furthermore, the experiment of HDNO was run on a Nvidia GeForce RTX 2080Ti graphic card, and it consumes around 3 hours for SL and 2 hours for RL. Therefore, it is not very expensive to reproduce our results.

C.2 Hyperparameters

The specific hyperparameters for training HDNO on MultiWoz 2.0 and MultiWoz 2.1 are shown in Table 8. During training, we used Adam optimizer (Kingma and Ba, 2015) for pretraining and

stochastic gradient descent (SGD) for RL. To ease life, we explicitly label the hyperparameter if it is only for pretraining or RL.

Table 8: Hyperparameters for training on MultiWoz 2.0 and MultiWoz 2.1.

Hyperparameters	MultiWoz 2.0	MultiWoz 2.1	Description
max_utt_len	50	50	The maximum length of a user's utterances.
max_dec_len	50	50	The maximum length of system's utterances in a response.
batch_size (Pretrain)	32	32	The number of samples for each update during SL.
learning rate (Pretrain)	1e-3	1e-3	The learning rate for SL.
grad_clip (Pretrain)	1.0	5.0	The maximum total norm of gradients is allowed during SL.
dropout (Pretrain)	0.5	0.5	The randomness coefficient of dropout.
num_epoch (Pretrain)	50	50	The number of training epochs for SL.
embed_size	100	100	The size of a word embedding vector.
utt_cell_size	300	300	The size of hidden layer for utterance encoder.
dec_cell_size	300	300	The size of hidden layer for natural language generator.
y_size	200	200	The size of a latent dialogue act.
beta (Pretrain)	1e-2	1e-3	The regularization coefficient of KL divergence for dialogue policy during SL.
eta (Pretrain)	0.1	0.1	The coefficient to control the contribution of randomly generated utterances.
high-level learning rate (RL)	9e-3	1e-2	The learning rate for high-level policy during RL.
low-level learning rate (RL)	9e-3	1e-2	The learning rate for low-level policy during RL.
num_epoch (RL)	1	2	The number of training epochs for RL.
temperature	0.1	0.1	The temperature of a Gibbs distribution for each word.
discount_factor (RL)	0.99	0.99	The importance of the future reward.
grad_clip (RL)	0.85	0.95	The maximum total norm of gradients is allowed during RL.
alpha (RL)	1e-4	1e-2	The multiplier to adjust the balance between the discriminator and the success rate in the total reward during RL.

D EXTRA BACKGROUND

D.1 TASK-ORIENTED DIALOGUE SYSTEM

A task-oriented dialogue system aims to help fulfill a user's task through conversation in turns. In general, each dialogue is modelled with an ontology called goal which includes inform slots and request slots. The traditional modular dialogue system is constituted of natural language understanding (NLU), dialogue state tracker (DST), dialogue policy and natural language generator (NLG). For a dialogue system, it needs to infer inform slots from user utterances and transform them to a dialogue state, which is completed by NLU and DST (Chen et al., 2017). In this work, we focus on optimizing dialogue policy and NLG, leveraging oracle dialogue states and database search results to produce dialogue acts and then responses (that should include as many request slots as possible) in turns. For optimizing dialogue policy, it is modelled with Markov decision process (MDP) (Williams and Young, 2007).

D.2 CHALLENGES OF TASK-ORIENTED DIALOGUE SYSTEM

We identify the main challenges of task-oriented dialogue system: (1) A dialogue with a single domain (i.e., completing one task in a dialogue) has been broadly studied, however, handling a dialogue with multiple domains is more challenging and needs more studies on it (Budzianowski et al., 2018); (2) If ignoring the syntactic structure of generated system utterances (i.e. losing comprehensibility), the mission of task-oriented dialogue can be simplified to generating corresponding labels (i.e., slots) for user utterances. Several existing algorithms already reached high scores on request slots acquisition but low scores on the comprehensibility of generated system utterances (Zhao et al., 2019; Mehri et al., 2019), so the simplified task is well-addressed. Reversely, if only focusing on the comprehensibility, the score on request slots acquisition could be drastically affected (Chen et al., 2019; Hosseini-Asl et al., 2020). In this work, we investigate the trade-off between the comprehensibility and request slots acquisition; (3) Designing and annotating a dialogue act structure is effort demanding (Budzianowski et al., 2018). Learning a meaningful latent dialogue act, therefore, becomes a new challenge (Zhao et al., 2019).

D.3 REINFORCEMENT LEARNING AND POLICY GRADIENT METHOD

MDP (Bellman, 1957) is a discrete time stochastic control process that an agent decides an action $\boldsymbol{a}_t \in \mathbb{A}$ at each state $\boldsymbol{s}_t \in S$ emitted from an environment via a probabilistic transition function $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t) \mapsto [0, 1]$. The environment also gives a reward $r(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}) \mapsto \mathbb{R}$ (abbreviated as r_t for simplicity) to measure the performance of an action at each time step. RL (Sutton and

Barto, 2018) is a learning paradigm which aims to find an optimal policy $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t) \mapsto [0,1]$ to deal with MDP by maximizing the expectation over cumulative long-term rewards. Mathematically, it can be expressed as $\max_\pi \mathbb{E}_\pi[\ \sum_{t=0}^{\infty} \gamma^t\ r_t\ ]$, where $\gamma \in (0,1)$ is a discount factor. Different from value-based methods, the policy gradient method derives a stochastic policy directly by optimizing a performance function w.r.t. parameters of the policy (Sutton and Barto, 2018). However, since the performance function cannot be differentiated w.r.t parameters of the policy, the policy gradient is derived by a natural gradient such that $\nabla_\theta J(\theta) = \mathbb{E}_\pi[\ Q(\boldsymbol{s}_t, \boldsymbol{a}_t)\nabla_\theta \ln \pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)\ ]$. REINFORCE (Williams, 1992) is a policy gradient method that evaluates $Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$ by a return $G_t = \sum_{t=0}^{\infty} \gamma^t\ r_t$. To deal with a continuous action space, $\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)$ can be represented as a Gaussian distribution (Sutton and Barto, 2018), where the mean and scale are both parameterized with θ.

E Dialogue Act represented as Ontology

Table 9: Dialogue act ontology.

Dialogue Act Type	inform / request / select / recommend / not found / request booking info / offer booking / inform booked / decline booking / welcome / greet / bye / reqmore

In this section, we show the ontology for representing a handcrafted dialogue act (Budzianowski et al., 2018) in Table 9. The semantic meanings of latent dialogue acts analyzed in Section 5.2.3 in the main part of paper are correlated to these dialogue act types. This is the reason why we conclude that learning latent dialogue acts can potentially substitute for handcrafted dialogue acts with ontology.

F Extra Discussion

Discussion on Reinforcement Learning for Task-oriented Dialogue System: As we stated in Section 3 in the main part of paper, the primary issue of using reinforcement learning (RL) in task-oriented dialogue system is that the improvement on fulfilling user requests and comprehensibility of generated system utterances is not simple to be balanced. One reason could be that the reward is only set up to improve the success rate and during learning the aspect of comprehensibility may be easily ignored. This is the reason why we consider extra reward criteria in our work to mitigate this predicament. Another reason could be that the state space and action space are so large in an end-to-end (E2E) model so that learning a mapping between these two spaces becomes difficult, as stated in Section 1 in the main part of paper. To deal with it, we propose to decouple dialogue policy and natural language generator (NLG) of an E2E model into two separate modules as those in the traditional architecture during learning, as well as model them with the option framework (Sutton et al., 1999). As a result, the complexity of mapping from context to system utterances is reduced from V^2 to $(L+M)V + ML$, where V is the vocabulary size, $L \ll V$ is the space size of latent dialogue acts and $M \ll V$ is the space size of encoded utterances.