Task-Optimized Adapters
for an End-to-End Task-Oriented Dialogue System

Namo Bang[*]   **Jeehyun Lee**[*]   **Myoung-Wan Koo**
Department of Artificial Intelligence, Sogang University, Korea
{namo950815, jhlee22, mwkoo}@sogang.ac.kr

Abstract

Task-Oriented Dialogue (TOD) systems are designed to carry out specific tasks by tracking dialogue states and generating appropriate responses to help users achieve defined goals. Recently, end-to-end dialogue models pre-trained based on large datasets have shown promising performance in the conversational system. However, they share the same parameters to train tasks of the dialogue system (NLU, DST, NLG), so debugging each task is challenging. Also, they require a lot of effort to fine-tune large parameters to create a task-oriented chatbot, making it difficult for non-experts to handle. Therefore, we intend to train relatively lightweight and fast models compared to PLM. In this paper, we propose an End-to-end TOD system with Task-Optimized Adapters which learn independently per task, adding only small number of parameters after fixed layers of pre-trained network. We also enhance the performance of the DST and NLG modules through reinforcement learning, overcoming the learning curve that has lacked at the adapter learning and enabling the natural and consistent response generation that is appropriate for the goal. Our method is a model-agnostic approach and does not require prompt-tuning as only input data without a prompt. As results of the experiment, our method shows competitive performance on the MultiWOZ benchmark compared to the existing end-to-end models. In particular, we attain state-of-the-art performance on the DST task of 2.2 dataset.[1]

1 Introduction

Task-oriented dialogue systems are trained to achieve specific goal to enhance efficiency and convenience in various fields such as customer service centers and healthcare information retrieval.

Task-oriented dialogue systems are divided into key components: understanding the user's intent (NLU), tracking the current dialogue states (DST), and generating responses based on previous sessions (NLG). Pipeline-based systems separately train each component, so they have the advantage of optimizing each module and raising the performance of a given task. User feedback is, however, difficult to propagate to each module, and inputs to the component is dependent on the result of the previous module (Chen et al., 2017). Recently, dialogue systems have been trained in an end-to-end manner with transfer learning or pre-training networks with large dialogue corpora. However, building efficient end-to-end TOD systems requires a large amount of data and has some limitations due to parameter sharing. End-to-end models back-propagate to transfer the gradients of the output and end back to the entire neural network. They pose an issue of parameter efficiency as updating all parameters for every downstream scenario. Also, it is challenging to debug each task and take task-flow characteristics into account.

Therefore, we propose a simple structure of adding adapters to the core modules (NLU, DST, NLG) of the TOD system as shown in Figure 1. By using the adapters, it is possible to optimize each task with only a small amount of training parameters, remaining the pre-trained model's parameters fixed. Additionally, it is safe from the catastrophic forgetting problem (French, 1999), which causes pre-trained models to lose important skills acquired during the pre-training process. The key is to apply a transfer learning strategy that yields compact and extensible downstream models in the dialogue system (Houlsby et al., 2019). This makes it easy for people to train large-scale end-to-end TOD models. Also, by applying REINFORCE (Sutton et al., 1999), we attempt to reduce the expected score gap caused by the small parameters of adapter compared to the full fine-tuning. Specifically, we use
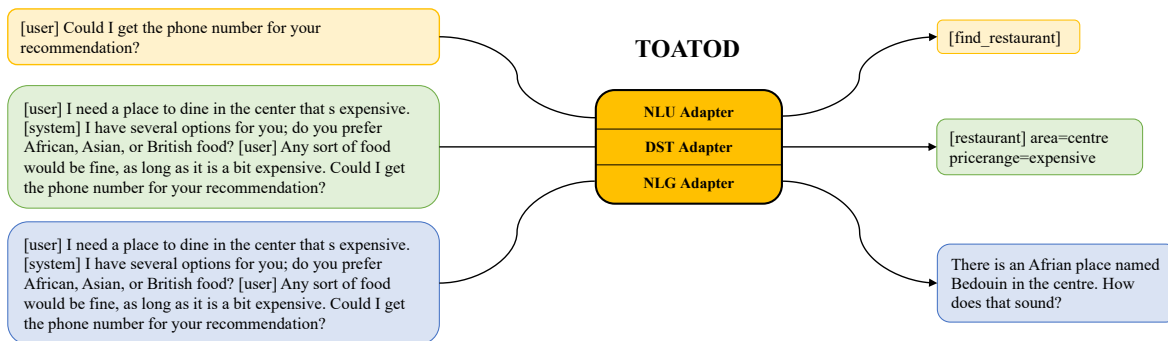

Figure 1: Overview of the Task-Optimized Adapters for an End-to-End Task-Oriented Dialogue System

Joint Goal Accuracy, and weighted sum of BLEU and Success rate as rewards for training DST and NLG adapter. To best of our knowledge, this is first work that use Joint Goal Accuracy as a reward for E2E TOD system.

To address the aforementioned problems, we propose a **T**ask-**O**ptimized **A**dapter for an end-to-end **T**ask-**O**riented **D**ialogue system (**TOATOD**) applying reinforcement learning to DST and NLG tasks. In summary, our key contributions are as follows:

- We present a new architecture that can debug per task of the end-to-end model using the separated adapters.

- Without updating the original parameters of PLM, we train end-to-end TOD system efficiently with a few trainable parameters.

- It is a novel approach to design a reward function not just for NLG, but also for DST task with metric-aware reinforcement learning.

- The performance of the proposed approach outperforms on the DST task of MultiWOZ 2.2 and shows comparable results to full fine-tuning on the NLU and NLG tasks.

2 Background

Pipeline-based Task-Oriented Dialogue System Conventional task-oriented dialogue systems usually based on the pipeline method, consisted of language understanding (NLU), dialogue state tracking (DST), policy learning (POL), and language generation (NLG). This kind of modularization allows for each component to be optimized independently, making it easier to update and understand how the model is working. Pipeline-based systems,

however, have several limitations. Each of the modules train sequentially, so it is hard to align modules to the common optimization targets (Liu and Lane, 2018). This makes the system more complex and harder to backpropagate cumulated errors. The performance of the previous components affects the next modules, so if upper modules perform poorly, errors that occurred earlier may propagate and be amplified in downstream components (Liu and Lane, 2018).

End-to-end Task-Oriented Dialogue System End-to-end task-oriented dialogue systems, on the other hand, are easier to optimize and train to directly map the input to output in a single model. They can leverage large amounts of data for robust learning and the entire system can be optimized under end-to-end settings, which leads to better performance. A general approach for building end-to-end systems is to fine-tune pre-trained language models (Budzianowski and Vulić, 2019). This approach utilizes the strength of pre-trained networks, which help the models to leverage the pre-trained knowledge while also adapting to task-specific data. For example, SimpleTOD (Hosseini-Asl et al., 2020) solved task-oriented dialogue as causal language modeling task using several versions of GPT (Radford and Narasimhan, 2018; Radford et al.).

Pre-training of Dialogue Language Model Recently, methods with pre-training dialogue language model (Wu et al., 2020; Zhang et al., 2020b; Peng et al., 2021; Su et al., 2022; He et al., 2022a), instead of fine-tuning pre-trained networks have outperformed the previous baselines on the benchmark. For instance, SPACE-3 (He et al., 2022a) captures the contextualized knowledge from large-scale dialogue corpora by pre-training the unified language model. However, there are still some issues with these methods. A large amount of param-
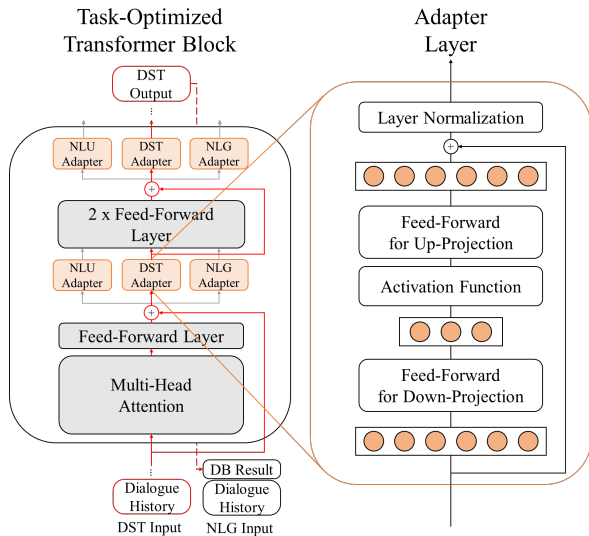
Figure 2: Architecture of the sub-modules in TOATOD. A red line indicates the forward path during DST inference. The DB result, based on the output of DST inference, is used for NLG inference sequentially. All of the adapter layers (NLU, DST, NLG) share the same architecture.

eters is required for training backbone models like BERT, T5 (Raffel et al., 2020), GPT, and UniLM (Dong et al., 2019). As shown in Table 1, T5$_{base}$ and T5$_{small}$ require the trainable parameters over 220M and 60M respectively. And they disregard the task-flow features of task-oriented dialogue systems. Also, it's still hard to debug per module because the model parameters are shared and jointly optimized. PPTOD (Su et al., 2022) integrated modules into a unified model with task-specific prompts and alleviated the error accumulation in plug-and-play way. Still, this method is not completely free from the interference among tasks due to fully shared parameters.

Adapter tuning for NLP Since the pre-trained models for NLP tasks have become mainstream, they are mainly used for transfer learning downstream tasks. However, a parameter efficiency issue has been raised because updating all PLM parameters is expensive for every downstream scenario. To address the issue, the adapter module is proposed to transfer PLM like BERT with parameter-efficient tuning (Houlsby et al., 2019) and shows comparable performance to full fine-tuning. The main idea behind the adapter is to train the network to the downstream task with task-specific parameters while maintaining the original pre-trained parameters. The module is composed of two feed-forward layers and a non-linear layer, which can be

inserted into the transformer blocks of an end-to-end model. It projects d-dimensional input features into a smaller dimension m, and then projects back into the original dimension, so the total of parameters add per layer with biases is $2md + d + m$. The number of additional parameters per task can be restricted by setting $m < d$ (Li et al., 2021). In most of the previous research, the adapter was used for efficient learning. But in this approach, we adapt the adapter modules not just for efficient learning (Stickland and Murray, 2019), but also task-optimized learning as recent studies that have mainly focused on separating parameters (Lin et al., 2021; Feng et al., 2022; Bapna and Firat, 2019).

Reinforcement learning for Text Generation Although token-based supervised learning is a widely adopted training method in text generation tasks, as highlighted by Ranzato et al., 2016, there are two major problems associated with this approach. The first problem is the exposure bias problem, where during training, the model is exposed to the ground-truth outputs, thus allowing it to learn to generate text that is similar to the training data. However, during evaluation, the model is not exposed to the ground-truth outputs and instead, generates texts based on the previous words generated by the model itself. This can lead to errors and a significant deviation in the generated text from the training data. The second problem is the token-level loss problem, where the training objective function is based solely on individual words' prediction, neglecting the predicted token sequence's overall coherence and fluency.

To address these problems, researchers have applied reinforcement learning to text generation tasks (Ranzato et al., 2016; Li et al., 2016; Paulus et al., 2018; Chen et al., 2020; Wang et al., 2021; Ye et al., 2022) using metrics such as BLEU (Papineni et al., 2002) or ROUGE (Lin, 2004) as rewards for sequence-level training. In our study, we also use the REINFORCE method and task-oriented dialogue metrics to continually train our model after supervised learning, not only to address these challenges but also to mitigate performance degradation caused by the use of an Adapter.

3 Methodology

3.1 Adapter for each task (NLU, DST, NLG)

The Adapter module is designed to adapt the pre-trained network for each of the tasks (NLU, DST,

	Pre-trained		Trainable per task	
Model	T5$_{base}$	T5$_{small}$	TOATOD$_{base}$	TOATOD$_{small}$
# of Prams	220M	60M	36M (14%)	7.9M (12%)

Table 1: This table shows the size of the pre-trained T5 model (frozen shared parameters) and trainable Adapter per task in our models. We do not experiment with a large model, because the trainable parameter size of TOATOD$_{large}$ is bigger than T5$_{base}$.

NLG) in a task-oriented dialogue system. As illustrated in Figure 2, the adapters are inserted after the feed-forward layer following the multi-head attention layer of the transformer blocks of the end-to-end model. It enables the model to learn task-specific representations while preserving the shared parameters learned during pre-training. As described in Table 1, our model consists of large-size frozen shared parameters per task and small size of trainable parameters that account for about 14% of the entire network. While original network's parameters are frozen, the jth adapter of task $i \in$ {NLU, DST, NLG}, A_{ij} computes as below:

$$A_{ij} = LN \left(W_{up} * ReLU \left(W_{down} * H_j \right) + H_j \right) \quad (1)$$

The output of the jth feed-forward layer with residual connection in the transformer block is represented as $H_j \in \mathrm{R}^{n \times d}$, where n is the input dimension, and d is the hidden dimension. As shown in Figure 2, the overall architecture of the adapter module includes multiple feed-forward projections, referred to as down-projection and up-projection, followed by layer normalization (LN). The down-projection with $W_{down} \in \mathrm{R}^{d \times h}$ projects the input H_j, which is passed by the ReLU activation function, and then the up-projection with $W_{up} \in \mathrm{R}^{h \times d}$ projects the output back to the original dimension. The bottleneck dimension h is a hyperparameter to project the original input to a smaller dimension. And each adapter has a residual connection to avoid vanishing gradient (Rebuffi et al., 2017; He et al., 2016).

3.2 Metric-Aware Reinforcement Learning for DST & NLG module

The overall loss function for the metric-aware reinforcement learning is given by the equation (2):

$$J(\theta) = \alpha \times J_{policy}(\theta) + (1 - \alpha) \times CE(y, \hat{y}) \quad (2)$$

The equation describes how the network updates the parameters in order to maximize both the likelihood of the generated response (token loss) and the quality of the response (policy loss). The token loss can be described as a categorical cross-entropy loss. $\hat{y}$ denotes the predicted probability of ground truth and y is the target probability. The token loss $CE(y, \hat{y})$, which measures how well a set of predicted token probabilities match the target tokens for a given context of dialogue when reinforcement learning. By applying REINFORCE method, the network can update weights towards the direction that allows the model for getting more rewards even when the reward function is non-differentiable.

$$J_{policy}(\theta) = -logP(\hat{y}) \times Reward(y, \hat{y}) \quad (3)$$

The policy loss $J_{policy}(\theta)$ is introduced to measure of how well the model can generate a token sequence with high probabilities that result in high rewards. $\hat{y}$ denotes the predicted token sequence by model. The policy loss described in the equation (3) is calculated as the negative log probability of the token sequence that has the highest probability multiplied by the reward.

The parameter α in the overall loss (2) is a hyperparameter, that is a scalar value between 0 and 1 to weigh the importance of these losses. In this way, the model is trained to predict the correct labels (categorical cross-entropy loss) and to make good decisions that result in high rewards (policy loss) at the same time. We define the reward functions of the DST and NLG modules as follows:

$$Reward_{DST} = JGA(y, \hat{y}) + 1 \quad (4)$$

The $Reward_{DST}$ is calculated as the sum of the Joint Goal Accuracy (JGA) and a constant value of 1. The JGA measures how well the model predicted the values for every slot in the dialogue turns. Using JGA as a reward, the model is encouraged to accurately track the state of the dialogue, which is crucial for generating appropriate responses and improving the performance of task-oriented dialogue systems.

$$Reward_{NLG} = (1 - \beta) \times \mathbb{E}[BLEU(y_u, \hat{y_u})] + \beta \times Success(y, \hat{y}) + 1 \quad (5)$$

In the equation (5), BLEU score and Success rate are used as rewards to guide the learning process

of NLG module. y_u denotes the ground truth token sequence of each utterance, and $\hat{y}_u$ denotes the predicted token sequence. To calculate the success rate, we apply the batch of a session-level, not an utterance-level. y and $\hat{y}$ without u mean the session level ground truth and prediction. The weighting factor β is adjusted to balance these two metrics. The hyperparameter β may have to be carefully chosen because there is a trade-off relationship between these two metrics in the RL setting (Wu et al., 2021), where increasing one metric may come at the cost of decreasing the other. We experiment to choose α and β on the Section 6.1.3 and 6.1.4.

4 Experimental Setup

4.1 Datasets

We experiment our method for dialogue state tracking (DST) and end-to-end response generation (NLG) tasks on the MultiWOZ 2.1 and 2.2 datasets, and the intent prediction (NLU) task with the Banking77, CLINC150, and HWU64 datasets.

MultiWOZ - 2.1, 2.2 The MultiWOZ (Budzianowski et al., 2018) dataset has been widely used to evaluate the performance of TOD systems. It consists of 8438, 1000, and 1000 for training, dev, test sets with multi-turn dialogues, collected through a Wizard-of-Oz (WOZ) setup. The dialogues cover a wide range of domains and topics. MultiWOZ 2.2 (Zang et al., 2020) is the improved version of MultiWOZ 2.1 (Eric et al., 2020) that has corrected annotation errors, inconsistencies, and ontology issues, also added span annotations to standardize.

Banking 77 (Casanueva et al., 2020) This dataset is a collection of 77 real-life customer banking service queries. It consists of 13,083 utterances. Each query is labeled with a single intent, however, it is hard to differentiate because they correspond to very similar tasks.

CLINC150 (Larson et al., 2019) This dataset is multi-domain dataset which contains 23,700 utterances that cover 150 intent classes over 10 domains.

HWU64 (Liu et al., 2019) This dataset consists of 25,716 examples. It maps user utterances to structured, but mode abstract. The data provides annotation with the 64 intents from 21 different domains.

Model	Backbone Model (Trainable Prams)	MultiWOZ 2.1	MultiWOZ 2.2
TRADE	-	46.0	45.4
DS-DST	BERT$_{base}$ (110M)	51.2	51.7
DST-Picklist	BERT$_{base}$ (110M)	53.3	
TripPy	BERT$_{base}$ (110M)	55.3	
ConvBERT +DG+Multi	BERT$_{base}$ (110M)	58.7	
Trippy +SaCLog	BERT$_{base}$ (110M)	60.61	
SimpleTOD	DistilGPT-2 (82M)	56.45	
UniLM	UniLM (340M)	54.25*	54.25*
AG-DST	PLATO-2 (310M)	57.26	57.26
SPACE-3	UniLM (340M)	57.50	57.50
D3ST$_{base}$	T5$_{base}$ (220M)	54.2	56.1
D3ST$_{large}$	T5$_{large}$ (770M)	54.5	54.2
SDP-DST	T5$_{base}$ (220M)	56.66	57.60
PPTOD$_{base}$	T5$_{base}$ (220M)	57.10	
PPTOD$_{large}$	T5$_{large}$ (770M)	57.45	
D3ST$_{XXL}$	T5$_{XXL}$ (11B)	**57.80**	58.7
TOATOD$_{small}$	T5$_{small}$ (7.9M)	53.02	**61.92**
TOATOD$_{base}$	T5$_{base}$ (36M)	54.97	**63.79**

Table 2: Joint Goal Accuracy for DST results. We use the best models after hyperparameter tuning and applying REINFORCE. Our models get the slightly higher score on the 2.2 dataset. Trainable Params denotes the number of trainable parameters of the network. The values with * are from SPACE-3 (He et al., 2022a).

4.2 Baselines & Settings

For the DST task, we compare our models with other strong baselines including TRADE (Wu et al., 2019), DS-DST (Zhang et al., 2020a), DST-Picklist (Zhang et al., 2020a), TripPy (Heck et al., 2020), ConvBERT+DG+Multi (Mehri et al., 2020a), TripPy+SaCLog (Dai et al., 2021), SimpleTOD (Hosseini-Asl et al., 2020), AG-DST (Tian et al., 2021), UniLM, SPACE-3, D3ST (Zhao et al., 2022), SDP-DST (Lee et al., 2021), and PPTOD. On the NLG task, we choose models trained on PLM in an end-to-end setting such as DoTS (Jeon and Lee, 2021), PPTOD, UBAR (Yang et al., 2021), MTTOD (Lee, 2021), RSTOD (Cholakov and Kolev, 2022), GALAXY (He et al., 2022b), MinTL (Lin et al., 2020), SOLOLIST, BORT (Sun et al., 2022a) and Mars (Sun et al., 2022b). For the NLU task, we compare TOATOD with existing baselines of each dataset.

We use T5$_{small}$ and T5$_{base}$ as backbone models initialized with PPTOD (Su et al., 2022)'s pretrained weights, which are trained on a large dialogue dataset. As described in the Appendix A, the bottleneck dimension h of the adapter is 1/2 of

Model	Backbone Model (Trainable Params)	MultiWOZ 2.1				MultiWOZ 2.2			
		Inform	Success	BLEU	Combined	Inform	Success	BLEU	Combined
DoTS	$BERT_{base}$ (110M)	86.65	74.18	15.90	96.32	-	-	-	-
PPTOD	$T5_{base}$ (220M)	87.09	79.08	19.17	102.26	-	-	-	-
UBAR	GPT2 (1.5B)	95.70	81.80	16.50	104.94	83.4	70.3	17.6	94.4
MTTOD	$T5_{base}$ (360.9M)*	90.99	82.08	19.68	106.22	85.9	76.5	19.0	100.2
RSTOD	$T5_{small}$ (105.5M)*	93.50*	84.70*	19.24*	108.34*	83.5	75.0	18.0	97.3
GALAXY	UniLM (340M)	95.30	86.20	**20.01**	**110.76**	85.4	75.7	19.64	100.2
MinTL	$BART_{large}$ (440M)	-	-	-	-	73.7	65.4	19.4	89.0
SOLOIST	GPT2 (1.5B)	-	-	-	-	82.3	72.4	13.6	90.9
BORT	$T5_{small}$ (60M)	-	-	-	-	85.5	77.4	17.9	99.4
Mars	$T5_{small}$ (60M)	-	-	-	-	88.9	78.0	**19.9**	**103.4**
$TOATOD_{small}$	$T5_{small}$ (7.9M)	92.10	80.40	18.29	104.54	85.80	74.00	18.00	97.90
$TOATOD_{base}$	$T5_{base}$ (36M)	**97.00**	**87.40**	17.12	109.32	**90.00**	**79.80**	17.04	101.94

Table 3: Inform, Success, BLEU, Combined Score for NLG results. All results of other models are cited from the official leaderboard. The values with * are from RSTOD. For the MultiWOZ 2.2 evaluation, we used our models trained on the MultiWOZ 2.1 after replacing the DST-optimized adapter with those trained on MultiWOZ 2.2.

the hidden dimension of the T5 model. We use the Adafactor (Shazeer and Stern, 2018) optimizer with 15 epochs and set batch size as 16, learning rate of 1e-4 during supervised learning of DST and NLG tasks. We sweep a wide range of learning rates: {1e-5, 1e-6, 1e-7}. For reinforcement learning, we train models 10 epochs for DST and 3 epochs for NLG. We do not train NLG-optimized adapters for the MultiWOZ 2.2, because there are not significant changes of response annotations from MultiWOZ 2.1 and intend to train robust models with noised dataset.

We follow the preprocessing method from UBAR to delexicalize slot values for each system responses. We evaluate our models using the older version of the standardized evaluation script for MultiWOZ 2.1, and the newly opened version for MultiWOZ 2.2, released by Nekvinda and Dusek, 2021. It has been adopted by the official MultiWoZ dataset github [2]. Other implementation details are described in Appendix C.

5 Experimental Results

5.1 Dialogue State Tracking

We evaluate our models on the DST task with the MultiWOZ 2.1 & 2.2 datasets. We compute Joint Goal Accuracy on the test set, which measures how many values are filled accurately compared to the ground truth states for all slots. Joint Goal Accuracy is considered as more difficult and important metric in most research (Zhou and Small, 2019;

---

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

Dey et al., 2022), because once wrong prediction has been made, it cannot get the score at that turn.

5.1.1 Evaluation Result

We compare our best models, $TOATOD_{small}$ and $TOATOD_{base}$ to the models trained with a pre-trained network. Table 2 shows that our models are competitive to the end-to-end models on the current benchmark. In the 2.1 dataset, our models show a relatively good performance, despite the small number of trainable parameters. As shown in the Table 5, the Joint Goal Accuracy of $TOATOD_{base}$ only with task-optimized adapter (SL) is 53.33, which is slightly lower than other models using $T5_{base}$ as backbone. So, we reduce the performance degradation applying metric-aware REINFORCE and report the final results. Trainable parameters of our models are less than 1/2 of the models with the smallest parameters. The result implies that the adapter module helps the network more adaptable to the DST task by only activating a few parts of the model. Because of relatively small parameters, our model is more robust to overfitting problem with confused labels. As mentioned in Section 4.1, MultiWOZ 2.2 is the cleaned version of 2.1 dataset, so the performance is better in the 2.2. Among the top results on the 2.2 dataset, our models obtain state-of-the-art performance. It demonstrates that TOATOD optimizes well on the given task remaining the prior knowledge learned from the pre-trained network.

5.2 End-to-End Response Generation

We test our methods with end-to-end response generation (NLG) task on the MultiWOZ 2.1 & 2.2 as in DST evaluation. Four metrics are used to measure the quality of generated responses. We measure if the system provides the appropriate entity (Inform rate), answers all the requested information (Success rate), and responds fluently (BLEU score). And the Combined Score for end-to-end response generation is computed as 'BLEU+0.5 × (Inform+Success)'. Under the end-to-end settings, the models have to predict proper dialogue states and then generate responses based on the states.

5.2.1 Evaluation Result

From Table 3, our models achieve comparable results (1.x point different from SOTA model) in all datasets. The adapter module helps each part of the model be fine-tuned independently, therefore we can optimize the DST and NLG task respectively. Our model performs well on NLG task based on the belief states from DST modules and base knowledge gained from the large-scale dialog dataset during pre-training. TOATOD$_{base}$ attains the best score on the Inform and Success rate. It shows that the reinforcement learning of our approach is effective for adjusting trade-off problem between BLEU score and others.

5.3 Intent Classification

Model	Banking77	CLINC150	HWU64
BERT-FIXED	87.19'	91.79'	85.77'
BERT-DG	91.75*	95.98*	90.89*
cist-dial (mslm)	91.17*	95.80*	91.36*
USE	92.81'	95.06'	91.25'
USE+CONVERT	93.36'	97.16'	92.62'
ConvBERT+Pre+Multi	93.44*	92.38*	**97.11***
SPACE 2.0	**94.77***	97.80*	94.33*
TOATOD$_{small}$	92.40	**98.45**	90.42
TOATOD$_{base}$	92.17	98.01	90.79

Table 4: Accuracy score (%) on all three NLU task dataset with full training. The values with ' are from banking77 paper (Casanueva et al., 2020), and * are from the leaderboard for DialoGLUE paper (Mehri et al., 2020b) and benchmark [3].

We test our models on the NLU task with Banking77, CLINC150, and HWU64. Intent prediction is the task to identify the intent behind a given in-

---

³ https://eval.ai/web/challenges/challenge-page/708/leaderboard/1943

put. The task is normally framed as a classification problem, so we set the metric as turn accuracy.

5.3.1 Evaluation Result

From Table 4, while our models do not achieve the highest score on NLU task, it is important to note that they perform well in relation to the size when compared to other models. This highlights the effectiveness of our task-optimized adapter approach in achieving a balance between model performance and efficiency.

6 Further Analysis and Discussion

6.1 Reinforcement Learning

6.1.1 w/o Reinforcement Learning

Task	DST		NLG			
Metrics	JGA	Slot F1	Inform	Success	BLEU	Combined
SL (2.2)	62.92	93.72	85.30	77.00	**18.44**	99.59
RL (2.2)	**63.79**	**93.96**	**90.00**	**79.80**	17.04	**101.94**
SL (2.1)	53.33	91.68	88.90	81.40	**18.73**	103.88
RL (2.1)	**54.97**	**92.01**	**97.00**	**87.40**	17.12	**109.32**

Table 5: Task performance of TOATOD$_{base}$ before and after applying REINFORCE. SL means supervised learning and RL means reinforcement learning. The test results of the TOATOD$_{small}$ is attached to Appendix B.

Described on the Table 5, after applying reinforcement learning, the performance of DST and NLG modules are enhanced. Models with reinforcement learning obtain best JGA on the DST task and get the highest Combined Score on the NLG task. The BLEU scores fall slightly but the Combined Score rise with Inform and Success, which means that incorporating reinforcement learning into the training process leads our model to complete tasks more efficiently. We conduct hyperparameter tuning only for the TOATOD$_{base}$, so there is some gap in the degree of performance improvement between TOATOD$_{small}$ and TOATOD$_{base}$.

6.1.2 Hyperparameters of REINFORCE

The hyperparameter α is used for balancing the importance between cross-entropy loss and policy loss. We re-train DST and NLG modules by optimizing the mixed objective function to get higher rewards. The second hyperparameter β is a scaling factor to control the trade-off between BLEU score

α	1.0	0.7	0.9	1.0	1.0	0.3	0.5	0.7	1.0
β	0.4	0.5	0.5	0.5	0.6	0.7	0.7	0.7	0.7
Inform	**97.50**	97.40	**97.50**	**97.50**	**97.50**	93.40	97.00	97.30	**97.50**
Success	85.70	86.40	85.90	85.70	85.80	85.40	**87.40**	87.10	86.10
BLEU	16.01	16.35	16.07	16.03	16.04	**18.36**	17.12	16.52	16.10
Combined	107.67	108.25	107.77	107.63	107.69	107.76	**109.32**	108.72	107.90

Table 6: Hyperparameter experiment with α and β on the NLG task. We experiment TOATOD$_{base}$ on the MultiWOZ 2.1.

and Success rate for NLG module. The BLEU score is calculated based on the number of matching n-grams between the generated text and the reference text. A higher score indicates that the generated text is more similar to the label text. Success rate measures the proportion of dialogues, where the model successfully completes the task. A higher success rate means the model is better at achieving the user's goal of the dialogue. In some cases, however, the model may generate text that is close to the reference text (high BLEU score) but not relevant to the current dialogue (low Success rate). So, we aim to reduce the gap between two metrics via hyperparameter tuning.

6.1.3 α of DST-optimized adapter

α	1.0	0.9	0.7
JGA	**54.97**	54.96	54.76
Slot F1	92.01	**92.02**	91.95

Table 7: Hyperparameter experiment with α on the DST task. We test TOATOD$_{base}$ on the MultiWOZ 2.1.

In the Table 7, the result of α experiment on DST task indicates that 1.0 yielded the best performance, suggesting the significance of the policy loss in the overall loss function. By maximizing policy loss, the model is encouraged to make decisions that result in high rewards, which improves the performance of the DST task.

6.1.4 α and β of NLG-optimized adapter

We experiment with several combinations of α and β within {α: 0.3, 0.5, 0.7, 0.9, 1.0 / β: 0.4, 0.5, 0.6, 0.7}. Regardless of hyperparameters, performances improve after applying the reinforcement learning. The hyperparameters of α=0.5 and β=0.7, result the best Combined Score, which is 5.44 point higher than the performance of the supervised learning. As shown in the Figure 3, on the experiment
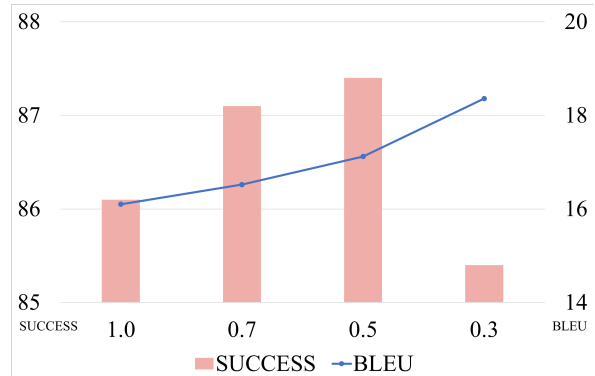

Figure 3: Effect of hyperparameter α when β is fixed (β=0.7). At the point of α is 0.5, trade-off issue occurs on MultiWOZ 2.1.

after removing impact of CE loss (α=1), we found that the higher the β, success rate increases compared to BLEU score as we expected.

As shown in Table 6, when the β is fixed, the bigger weight (smaller α) on the CE loss ensures the higher BLEU score. On the other hand, the success rate started to decline from the certain point. Trade-off issue appears at this point. Therefore, α and β need to be properly tuned in NLG task for good performance.

7 Conclusion

We propose TOATOD, task-optimized adapters for an end-to-end task dialogue system. By adapting task-optimized adapters, we utilize the end-to-end models without updating the pre-trained parameters and enabling debugging per task, which is different from previous research. In addition, we apply REINFORCE algorithm with metric-aware reward function directly not only on the NLG task but also DST task to prevent score degradation. As a result, we attain comparable performance to the previous SOTA models on every benchmark with very small number of trainable parameters. For the DST task of MultiWOZ2.2, our TOATOD model outperforms the current SOTA systems.

Limitations

We train the task-optimized adapters based on the pre-trained weights of dialogue LM. Therefore, if applied to other dialogue tasks such as chit-chat and conversational QA system, the performance could be lower than that shown in our research. And we need future works to clarify the reason why the performance was better on the MultiWOZ 2.2 dataset, which is expected that our model does not overfit to the confused labels. Our model inferences in on the end-to-end manner, but trains like modular system for each task. End-to-end learning is currently under study. We could adapt the multi-task end-to-end learning to our method, which may lead to the better performance. Also, we could analyze the inner working of task-optimized adapters applying XAI technologies.

Ethics Statement

We honor the ethical codes set out in the ACL code of Ethics. All of the datasets used in our study are from previous studies and do not have privacy issues.

Acknowledgements

References

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's GPT-2 - how can I help you? towards the use of pre-trained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22, Hong Kong. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. *CoRR*, abs/2003.04807.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems. *ACM SIGKDD Explorations Newsletter*, 19.

Yu Chen, Lingfei Wu, and Mohammed J. Zaki. 2020. Reinforcement learning based graph-to-sequence model for natural question generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Radostin Cholakov and Todor Kolev. 2022. Efficient task-oriented dialogue systems with response selection as an auxiliary task. *CoRR*, abs/2208.07097.

Yinpei Dai, Hangyu Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si, and Xiaodan Zhu. 2021. Preview, attend and review: Schema-aware curriculum learning for multi-domain dialogue state tracking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 879–885. Association for Computational Linguistics.

Suvodip Dey, Ramamohan Kummara, and Maunendra Desarkar. 2022. Towards fair evaluation of dialogue state tracking by flexible incorporation of turn-level performances. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 318–324, Dublin, Ireland. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. 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 13042–13054.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.

Shaoxiong Feng, Xuancheng Ren, Kan Li, and Xu Sun. 2022. Hierarchical inductive transfer for continual dialogue learning. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 693–699. Association for Computational Linguistics.

Robert French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3:128–135.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Wanwei He, Yinpei Dai, Min Yang, Jian Sun, Fei Huang, Luo Si, and Yongbin Li. 2022a. Unified dialog model pre-training for task-oriented dialog understanding and generation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 187–200, New York, NY, USA. Association for Computing Machinery.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, Jian Sun, and Yongbin Li. 2022b. GALAXY: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10749–10757. AAAI Press.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. TripPy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting. 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*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Hyunmin Jeon and Gary Geunbae Lee. 2021. Domain state tracking for a simplified dialogue system. *CoRR*, abs/2103.06648.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue state tracking with a language model using schema-driven prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4937–4949. Association for Computational Linguistics.

Yohan Lee. 2021. Improving end-to-end task-oriented dialog system with a simple auxiliary task. In *Conference on Empirical Methods in Natural Language Processing*.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Pretrained language models for text generation: A survey. *CoRR*, abs/2105.10311.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zhaojiang Lin, Andrea Madotto, Yejin Bang, and Pascale Fung. 2021. The adapter-bot: All-in-one controllable conversational model. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 16081–16083. AAAI Press.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. Mintl: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3391–3405. Association for Computational Linguistics.

Bing Liu and Ian Lane. 2018. End-to-end learning of task-oriented dialogs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 67–73, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019. Benchmarking natural language understanding services for building conversational agents. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction - 10th International Workshop on Spoken Dialogue Systems, IWSDS 2019, Syracuse, Sicily, Italy, 24-26 April 2019*, volume 714 of *Lecture Notes in Electrical Engineering*, pages 165–183. Springer.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tür. 2020a. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *CoRR*, abs/2009.13570.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tür. 2020b. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *CoRR*, abs/2009.13570.

Tomás Nekvinda and Ondrej Dusek. 2021. Shades of bleu, flavours of success: The case of multiwoz. *CoRR*, abs/2106.05555.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. SOLOIST: building task bots at scale with transfer learning and machine teaching. *Trans. Assoc. Comput. Linguistics*, 9:907–824.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. 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 506–516.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Asa Cooper Stickland and Iain Murray. 2019. BERT and pals: Projected attention layers for efficient adaptation in multi-task learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5986–5995. PMLR.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. Multi-task pre-training for plug-and-play task-oriented dialogue system. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4661–4676, Dublin, Ireland. Association for Computational Linguistics.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2022a. BORT: back and denoising reconstruction for end-to-end task-oriented dialog. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2156–2170. Association for Computational Linguistics.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2022b. Mars: Semantic-aware contrastive learning for end-to-end task-oriented dialog. *CoRR*, abs/2210.08917.

Richard S. Sutton, David A. McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 1057–1063. The MIT Press.

Xin Tian, Liankai Huang, Yingzhan Lin, Siqi Bao, Huang He, Yunyi Yang, Hua Wu, Fan Wang, and Shuqi Sun. 2021. Amendable generation for dialogue state tracking. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 80–92, Online. Association for Computational Linguistics.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. 2021. Modelling hierarchical structure between dialogue policy and natural language generator with option framework for task-oriented dialogue system. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, Online. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. 2021. Alternating recurrent dialog model with large-scale pre-trained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1292–1301. Association for Computational Linguistics.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. UBAR: towards fully end-to-end task-oriented dialog system with GPT-2. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14230–14238. AAAI Press.

Chenchen Ye, Lizi Liao, Fuli Feng, Wei Ji, and Tat-Seng Chua. 2022. Structured and natural responses co-generation for conversational search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 155–164, New York, NY, USA. Association for Computing Machinery.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wang, Philip S. Yu, Richard Socher, and Caiming Xiong. 2020a. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics,*

SEM@COLING 2020, Barcelona, Spain (Online), December 12-13, 2020, pages 154–167. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020b. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 270–278. Association for Computational Linguistics.

Jeffrey Zhao, Raghav Gupta, Yuan Cao, Dian Yu, Mingqiu Wang, Harrison Lee, Abhinav Rastogi, Izhak Shafran, and Yonghui Wu. 2022. Description-driven task-oriented dialog modeling. *CoRR*, abs/2201.08904.

Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *CoRR*, abs/1911.06192.

Appendices

A   Units of Adapters

Dim	1/2	1/4	1/8
JGA	**51.59**	51.11	50.43
Slot f1	91.02	**91.07**	90.40
Trainable Params	7.9M	7.2M	6.6M

Table 8: Adapter units experiment results. We test different bottleneck dimension of adapter module on the DST task.

In this experiment, we evaluate the performance with several bottleneck dimensions, h = 256, 128, 64 with TOATOD$_{small}$, which is 1/2, 1/4, 1/8 size of T5$_{small}$'s embedding dimension of 512. The main focus of the experiment is to investigate the effect of the bottleneck dimension h of the adapter module on the performance of the model. To evaluate the performance of the model, we use the Joint Goal Accuracy for DST task. We keep the other hyperparameters constant across all the experiments, including learning rate of 1e-4 and evaluate the performance on the test set of MultiWOZ 2.1. We report the result on the Table 8.

The result implies that there is a trade-off between the bottleneck dimension and the performance of the model. As the bottleneck dimension increases, the performance of the model also improves. The best performance is achieved with a bottleneck dimension of 256, where the JGA is

51.59. It is important to carefully choose the bottle-neck dimension when using the adapter module in the task-oriented dialogue system. As described in the Table 1, the trainable parameters of our model with bottleneck dimension of 256 is still significantly smaller than the PLM's parameters, so we choose the size of 1/2.

B w/o Reinforcement Learning of TOATOD$_{small}$

Task	DST		NLG			
Metrics	JGA	Slot F1	Inform	Success	BLEU	Combined
SL (2.2)	61.29	93.46	78.80	69.50	**18.46**	92.61
RL (2.2)	**61.92**	**93.65**	**85.80**	**74.00**	18.00	**97.90**
SL (2.1)	52.58	91.31	84.30	74.40	**18.82**	98.17
RL (2.1)	**53.01**	**91.61**	**92.10**	**80.50**	18.28	**104.58**

Table 9: Task performance of TOATOD$_{small}$ before and after applying REINFORCE. SL means supervised learning and RL means reinforcement learning.

C Implementation Details

We used 8 A100 (80G) GPUs, but they were fully used during only reinforcement learning. During supervised learning, we used 4 GPUs. While reinforcement learning of the DST task-optimized adapters, we set learning rate as 1e-5, and batch size as 32 (utterance-level). On the contrary, while reinforcement training of the NLG task-optimized adapters, we set learning rate as 1e-6 and batch size as 4 (session-level). During the entire training process, we set random seed as 42. And for the NLU task, we used 1 RTX A5000 (24GB) GPU and trained models without reinforcement learning. We set batch size as 64 and single run with random seeds. When training T5$_{small}$ and T5$_{base}$ for the Banking 77 and CLINC150, we used learning rates of 0.001 and 0.15. And we used learning rates of 0.01 and 0.1 for the T5$_{small}$ and T5$_{base}$ with the HWU64.

A For every submission:

☑ **A1. Did you describe the limitations of your work?**
Limitations

☑ **A2. Did you discuss any potential risks of your work?**
Limitations

☑ **A3. Do the abstract and introduction summarize the paper's main claims?**
1

☒ **A4. Have you used AI writing assistants when working on this paper?**
Left blank.

B ☑ Did you use or create scientific artifacts?

3

☑ **B1. Did you cite the creators of artifacts you used?**
2, 3

☑ **B2. Did you discuss the license or terms for use and / or distribution of any artifacts?**
4

☑ **B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?**
2

☐ **B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?**
Not applicable. All of the datasets used in our study are from previous studies and do not have privacy issues.

☐ **B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?**
Not applicable. Left blank.

☑ **B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.**
4

C ☑ Did you run computational experiments?

4.2 & Appendix C

☑ **C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?**
Table 1 & Appendix C

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4.2 & 6 & Appendix C

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Appendix C

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4.2

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.