Hierarchical Inductive Transfer for Continual Dialogue Learning

Anonymous ACL submission

Abstract

Pre-trained models have achieved excellent performance on the dialogue task. However, for the continual increase of online chit-chat scenarios, directly fine-tuning these models for each of the new tasks not only explodes the capacity of the dialogue system on the embedded devices but also causes knowledge forgetting on pre-trained models and knowledge interference between diverse dialogue tasks. In this work, we propose a hierarchical inductive transfer framework to learn and deploy the dialogue skills continually and efficiently. First, we introduce the adapter module into pre-trained models for learning new dialogue tasks. As the only trainable module, it is beneficial for the dialogue system on the embedded devices to acquire new dialogue skills with negligible additional parameters. Then, for alleviating knowledge interference between tasks yet benefiting the regularization between them, we further design hierarchical inductive transfer that enables new tasks to use general knowledge in the base adapter without being misled by diverse knowledge in task-specific adapters. Empirical evaluation and analysis indicate that our framework obtains comparable performance under deployment-friendly model capacity.

1 Introduction

Neural dialogue models (Shang et al., 2015; Serban et al., 2016) have drawn increasing attention due to their high commercial value. Previous work usually makes efforts to improve the diversity and coherence of the responses (Li et al., 2016; Serban et al., 2017; Shen et al., 2018; Zhang et al., 2018a,c). Recently, a lot of work tries to enable models to express empathy (Zhou et al., 2018; Rashkin et al., 2019), be knowledgeable (Ghazvininejad et al., 2018; Dinan et al., 2019), and demonstrate consistent personalities (Qian et al., 2018; Zhang et al., 2018b, 2019). Specifically, the dialogue model is trained on a task-specific dataset to learn the corresponding conversation skill. However, with the increasing number of online chit-chat scenarios, the dialogue system is further expected to continually specialize in new tasks without sacrificing the performance on old tasks. Meanwhile, the dialogue system must keep its capacity as small as possible for the deployment on the computation resource–limited embedded devices.

Pre-trained models (Radford et al., 2018; Devlin et al., 2019) have successfully facilitated the learning of the downstream tasks in various fields. To address the challenge of continual dialogue learning, directly fine-tuning pre-trained models on each of the new dialogue tasks is a straightforward way to equip the dialogue system with new conversation skills continually. However, it explodes the capacity of the dialogue system because knowledge of new tasks need to be stored in new pre-trained models to avoid erasing knowledge of old tasks. A more advanced approach is to multi-task one pre-trained model on all old tasks and then fine-tune it on new tasks, which can alleviate the capacity problem and use general knowledge between old tasks to improve the model performance on new tasks (Smith et al., 2020). Nonetheless, these advantages come at the cost of performance decline on some old tasks due to knowledge interference between diverse tasks.

To tackle these problems, we propose a hierarchical inductive transfer framework to construct and deploy the dialogue system with fewer computational resources. The framework is inspired by the fact that the conversational skills are multi-layered, and while general skills, e.g., uttering fluent sentences, are necessary for all scenarios and the requisite for sophisticated skills, specialized skills, such as negotiating and debating, work for fewer occasions. In the hierarchy of conversational skills, the later skills could be efficiently built upon the former skills if they are well-learned. However,

considering it is difficult to determine the proper order of the skills and the skills needs for a dataset, we take the following practical approach.

We first introduce the adapter module, consisting of a small sub-net, into the pre-trained model. Each block of the pre-trained model is assigned two adapters inserted after the self-attention layer and the feed-forward layer. During training, adapters, as the only trainable parameters, learn knowledge of dialogue tasks, which avoids knowledge forgetting on pre-trained models and therefore keeps the capacity of the dialogue system almost constant as the number of dialogue tasks increases. Then, we separate the adapter into the base adapter and the task-specific adapter to avoid the performance decline of models on old tasks caused by knowledge interference between diverse tasks. The former is multi-tasked with old tasks to obtain general knowledge by regularization between diverse tasks, which facilitates the learning of new tasks. The latter is further fine-tuned on any dialogue task to learn the corresponding task-specific knowledge, which maintains the model performance on old tasks. Finally, the proposed framework significantly enhances the training efficiency due to the learning of dialogue tasks only being conducted via adapters.

2 Method

In this section, we first describe the vanilla adapter and how to apply it to the dialogue tasks and then present the hierarchical inductive transfer to learn general knowledge and task-specific knowledge.

2.1 Adapter for Continual Dialogue Learning

Directly fine-tuning pre-trained models for each of the new dialogue tasks will cause knowledge forgetting, and therefore each task requires a large set of parameters for maintaining the model performance on both old and new tasks. Compared with it, we keep the parameters of the pre-trained model fixed and use the adapter to learn new tasks. Adapters are inserted after the self-attention layer and the feed-forward layer of each block of the pre-trained model, illustrated in Figure 1:

$$\mathbf{h}^{l+1} = \mathrm{LN}\left(\mathbf{h}^{l} + \mathrm{Ada}\left(\mathrm{Fun}\left(\mathbf{h}^{l}\right)\right)\right), \quad (1)$$

where $\mathbf{h}^{l}$ and $\mathbf{h}^{l+1}$ represent the input and the output of sub-blocks, and $\mathrm{Fun}(\cdot)$, $\mathrm{Ada}(\cdot)$, and $\mathrm{LN}(\cdot)$ represent the function layer (i.e., the self-attention layer or the feed-forward layer), the adapter, and the layer norm, respectively.

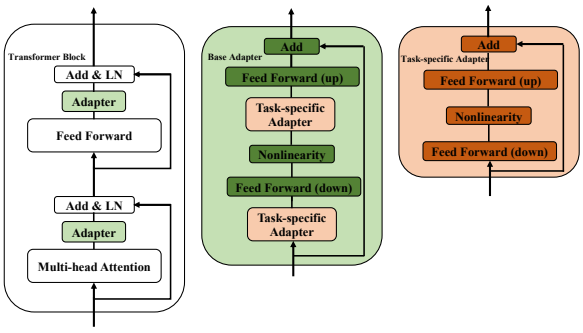

Figure 1: *An overview of the hierarchical inductive transfer framework.*

Each adapter consists of a bottleneck module with a skip-connection. Concretely, the bottleneck module first down-projects the d_o-dimension output of the previous layer into features with a smaller dimension, d_a, followed by a nonlinearity, and then up-projects to the original dimension. Formally, it can be expressed as:

$$\mathrm{Ada}(\mathbf{o}) = \mathbf{o} + W^{U} a\left(W^{D}\mathbf{o}\right), \quad (2)$$

where W^{D} $(d_o \times d_a)$ and W^{U} $(d_a \times d_o)$ are the parameters of the down- and the up-projections, and $a(\cdot)$ is the activation function. By adjusting the value of d_a, we can control the number of parameters of adapters to a deployment-friendly range.

For each new task, only a few parameters need to be trained on the cloud servers and delivered to the embedded devices, which significantly improves the training efficiency and reduces the size of the dialogue system. Please refer to Appendix B for a more detailed discussion.

2.2 Hierarchical Inductive Transfer

In continual dialogue learning, the old tasks usually contain useful knowledge for the learning of new tasks. But they may also have knowledge interference with new tasks. To alleviate this issue, one can multi-task the adapters with all old tasks and find general knowledge for new tasks. However, the regularization between diverse tasks also causes the performance decline of multi-tasked models on some old tasks due to knowledge interference among old tasks. Therefore, we further design a hierarchical inductive transfer framework that consists of two kinds of adapter, the base adapter and the task-specific adapter.

Specifically, we take the vanilla adapters as the base adapters and introduce a set of new adapters

Method	Θ	θ_Δ	ConvAI2	WoW	ED	BST	Average
FE	+ 0.216×	5.4 %	0.8698	0.9129	0.6255	0.7413	0.7874
FT	+ 4.0 ×	100 %	0.8855	0.917	0.6267	0.7838	0.8032
MT+FT	+ 2.0 ×	100 %	0.8878	**0.9274**	0.6241	**0.8241**	**0.8158**
Ada	+ 0.075×	1.87%	0.888	0.9177	0.6204	0.7662	0.7981
AdaHIT	+ 0.112×	4.2 %	**0.8914**	0.9193	**0.6358**	0.8167	**0.8158**

Table 1: Comparison in terms of total number of additional parameters (Θ), trainable parameters per task (θ_Δ), and performance on tasks. The proposed AdaHIT achieves performance competitive with the state-of-the-art (MT+FT) with far fewer total parameters to be stored and parameters to be trained.

inserted before the feed-forward layers of each base adapter as the task-specific sub-adapters, shown in Figure 1. It can be formulated as:

$$\mathrm{Ada}_{\mathrm{bs}}(\mathbf{o}) = \mathbf{o} + W^U \mathrm{Ada}_{\mathrm{ts}}\left(a\left(W^D \mathrm{Ada}_{\mathrm{ts}}(\mathbf{o})\right)\right), \quad (3)$$

where $\mathrm{Ada}_{\mathrm{bs}}(\cdot)$ and $\mathrm{Ada}_{\mathrm{ts}}(\cdot)$ represent the base adapter and the task-specific adapter. Each task-specific adapter also consists of a bottleneck module and a skip-connection.

During training, we first multi-task the base adapters with all old tasks to find general knowledge and then fine-tune a set of task-specific adapters for each task, including old tasks and new tasks, which enables the new task to benefit from the knowledge of old tasks without sacrificing the model performance on some old tasks.

3 Experiment

3.1 Datasets and Baselines

Datasets To evaluate the proposed framework, we take **ConvAI2** (an extension of the PersonaChat dataset (Zhang et al., 2018b)), Wizard of Wikipedia (**WoW**) (Dinan et al., 2019), Empathetic Dialogues (**ED**) (Rashkin et al., 2019), and Blended Skill Talk (**BST**) (Smith et al., 2020) as an instance of continual dialogue learning. The first three tasks are the old tasks and the last task represents the new task.

Baselines Four methods of inductive transfer are used to compare with our framework (**AdaHIT**), including feature extraction (**FE**), which adds and optimizes a classification layer on the top of the pre-trained model (Vaswani et al., 2017), fine-tuning (**FT**), which updates all parameters of the pre-trained model for each task, multi-tasking with fine-tuning (**MT+FT**), which first multi-tasks the entire pre-trained model with all old tasks and then fine-tunes it on the new task, and vanilla adapter (**Ada**), which trains a set of adapters for each task.

3.2 Experimental Settings

Following Smith et al. (2020), we use the poly-encoder with 256M parameters (Humeau et al., 2019) as the underlying model, pretrain it on the pushshift.io Reddit dataset, and then conduct inductive transfer on the downstream tasks. We also truncate the length of label and text to 72 and 360, and set the embedding size to 768 as Smith et al. (2020). The batch size is 128 and the other responses in a batch are set as negatives for training. The dimension of adapters d_a is 64. We adopt AdaMax (Kingma and Ba, 2015) as the optimizer throughout the experiments, and the learning rates are 9e-4, 2.5e-3, 1e-3, and 4e-4 for ConvAI2, WoW, ED, and BST. The total training epochs are 8 with linear warm-up for 10% and linear decay for the rest. All experiments are conducted using ParlAI[1].

3.3 Experimental Results

For the retrieval-based dialogue scenarios, we measure hits@1/K[2] on the validation set of each task for automatic evaluation. The number of candidates is 20 for ConvAI2 and 100 for other tasks. The results reported in Table 1 show that AdaHIT achieves the best average performance, the same as MT+FT, at the cost of far fewer parameters to be trained and stored, indicating the superiority of deployment on embedded devices. AdaHIT significantly outperforms Ada in both old tasks and new task with a slight regression of computational efficiency, which demonstrates that the hierarchical inductive transfer can extract general knowledge to facilitate the learning of the new task while boosting the model performance on old tasks effectively.

3.4 Ablation Study and Analysis

Effect of Base Adapter To analyze the effect of the base adapter, we train it with different tasks, and then test it on BST in a zero-shot manner, or a fine-tuning manner which is the same with AdaHIT. From the results in Table 2, we can observe that the base adapter with multi-tasking obtains the best performance under both the zero-shot

[1] https://parl.ai/

[2] hits@1/K represents recall@1 when choosing the gold response from K candidates.

Dataset for Adapter	BST (Zero-Shot)	BST (Fine-Tuning)
ConvAI2	0.753	0.8039
WoW	0.6222	0.7751
ED	0.6349	0.7846
MT	**0.768**	**0.8167**

Table 2: Effect of training datasets for the base adapter.

Number of Layers	1	2	3	4	5	6
AdaHIT	0.809	0.807	0.796	0.785	0.762	0.734
Position of Layers	0	2	4	6	8	10
AdaHIT	0.809	0.808	0.809	0.801	0.793	0.763

Table 3: Ablation Study in terms of number and position of removed adapters on BST.

and the fine-tuning setting, indicating that multi-tasking provides more general knowledge for the learning of BST. It is also worth mentioning that the base adapter trained on ConvAI2 achieves better performance than adapters on other tasks, because ConvAI2 contains more useful information, e.g., persona, that also exists every sample of BST.

Visualization To verify whether AdaHIT helps task adaption, we visualize the representations from models with different base adapters, i.e., trained on MT and ConvAI2, the result of which is shown in Figure 2. As we can see, the two models can both adjust to specific downstream tasks but representations with MT are better distributed and more tightly clustered. It is also interesting to see that the model with MT may implicitly distinguish the skills for each task, because while ED and ConvAI2 share more common skills, they are quite different from WoW, and such difference is evidently reflected by the visualization.

Ablation Study We further investigate the impact of adapters on model performance quantitatively. First, we gradually remove each trained adapter from the bottom layer, and then increase the number of removed adapters. As shown in Table 3, the adapters of higher layers have more significant effects than the adapters of lower layers, indicating that we can only insert the adapters into the higher layers to improve the training efficiency.

4 Related Work

Continual Dialogue Learning Neural dialogue models (Mou et al., 2016; Xing et al., 2017; Zhao et al., 2017) can acquire various kinds of conversation skills from corpora, such as characterizing personalities (Qian et al., 2018; Zhang et al., 2018b), expressing emotion and empathy (Zhou et al., 2018; Rashkin et al., 2019), and retrieving knowledge

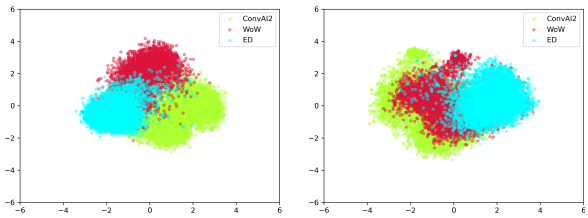

Figure 2: Visualization of learned sentence representations from AdaHIT with differently-trained base adapters. MT is on the left and ConvAI2 is on the right.

(Ghazvininejad et al., 2018; Dinan et al., 2019). Unlike existing work on enhancing a particular conversation skill, we work towards a new dialogue learning paradigm, where conversation skills are gradually embedded into a single model by mutual reinforcement instead of interference.

Inductive Transfer Continual learning in terms of transferring inductive knowledge from pre-trained models to downstream tasks can be categorized into feature-based, fine-tuning–based, and adapter-based (Ruder, 2019). We adopt the adapter-based approach that benefits from both pre-trained models and a small set of extra parameters for task-specific knowledge. Unlike conventional adapters (Houlsby et al., 2019; Poth et al., 2021; Pfeiffer et al., 2021), knowledge in the proposed adapters will be used to further boost the learning of new dialogue tasks, whereas knowledge of each task is separated into general and task-specific parts to avoid knowledge interference. Madotto et al. (2020) also uses the adapters to acquire the conversation skills, but it does not consider knowledge transfer and interference between adapters.

5 Conclusion

In this work, we propose a hierarchical inductive transfer framework to efficiently train and deploy the pre-trained models for growing numbers of new dialogue tasks requiring diverse skills. Considering the computation resource–limited embedded devices, we first adopt the adapter module, a small plug-in sub-net, as the only incremental and trainable parameters for learning each of the new dialogue tasks. To take advantage of knowledge in old tasks to facilitate the learning of new tasks, we further propose the hierarchical inductive transfer to alleviate knowledge interference between tasks and provide general knowledge for new tasks. Extensive experiments and analysis demonstrate that the proposed framework achieves high computational efficiency with competitive performance.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5110–5117. AAAI Press.

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.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Real-time inference in multi-sentence tasks with deep pretrained transformers. *CoRR*, abs/1905.01969.

Diederik P. Kingma and Jimmy Ba. 2015. 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*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics.

Andrea Madotto, Zhaojiang Lin, Yejin Bang, and Pascale Fung. 2020. The adapter-bot: All-in-one controllable conversational model. *CoRR*, abs/2008.12579.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 3349–3358. ACL.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. In *EACL*, pages 487–503. Association for Computational Linguistics.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to pre-train on? efficient intermediate task selection. In *EMNLP (1)*, pages 10585–10605. Association for Computational Linguistics.

Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Assigning personality/profile to a chatting machine for coherent conversation generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4279–4285. ijcai.org.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. 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 5370–5381. Association for Computational Linguistics.

Sebastian Ruder. 2019. *Neural transfer learning for natural language processing*. Ph.D. thesis, National University of Ireland, Galway.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3776–3784. AAAI Press.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3295–3301. AAAI Press.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation.

In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1577–1586. The Association for Computer Linguistics.

Xiaoyu Shen, Hui Su, Shuzi Niu, and Vera Demberg. 2018. Improving variational encoder-decoders in dialogue generation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5456–5463. AAAI Press.

Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2021–2030. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. 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 5998–6008.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3351–3357. AAAI Press.

Hainan Zhang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2018a. Reinforcing coherence for sequence to sequence model in dialogue generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4567–4573. ijcai.org.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018b. Personalizing dialogue agents: I have a dog, do you have pets too? 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 2204–2213. Association for Computational Linguistics.

Wei-Nan Zhang, Qingfu Zhu, Yifa Wang, Yanyan Zhao, and Ting Liu. 2019. Neural personalized response generation as domain adaptation. *World Wide Web*, 22(4):1427–1446.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018c. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 1815–1825.

Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 654–664. Association for Computational Linguistics.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 730–739. AAAI Press.

A Structure of Adapters

We have designed and evaluated diverse structures of adapters for continual dialogue tasks, such as the self-attention structure and the convolutional structure. However, there is no significant effect on performance, which is in line with previous adapter-based work. For the basic bottleneck structure, there are two advantages. First, it can limit the number of parameters per adapter by setting the bottleneck dimension $d_a \ll d_o$. Second, it also provides a flexible way to trade-off model performance with parameter efficiency.

B Training Efficiency of Adapters

Compared with the traditional fine-tuning method, our framework conducts the learning of dialogue tasks only by adapters, which reduces the memory requirements and the computing operations of each batch and therefore trains more samples with the same time. For example, there is a two-layer network, and only the first layer is trainable:

$$y_1 = f(w_1 * x + b_1)$$

$$y_2 = f(w_2 * y_1 + b_2)$$

Although we still need to calculate $\frac{\partial y_2}{\partial y_1}$ due to the chain rule, we do not calculate $\frac{\partial y_2}{\partial w_2}$ and $\frac{\partial y_2}{\partial b_2}$ (i.e., reducing the computing operations) and do not save them (i.e., reducing the memory requirements) for the parameter update. Considering the number of parameters of Transformer, the proposed framework indeed improves the training efficiency. Moreover, we can only insert the adapters in the top layers because the adapters in the bottom layers have a weaker effect on the model performance, indicated by Table 3, which limits the chain derivative to the top layers and further reduces the computing operations.

C Knowledge Forgetting of FT

In order to demonstrate knowledge forgetting of the traditional fine-tuning method, we evaluate the performance of the multi-tasking model (MT) on the old tasks before and after being fine-tuned on the new task. As shown in Table 4, the model fine-tuned on the new task (BST) shows consistent performance degradation on the old tasks.

For a fair comparison, both our method (AdaHIT) and MT+FT are multi-tasked on the old tasks and then fine-tuned on the new task. We also provide the results of a stronger model that is first pre-trained on the old tasks and then multi-tasked on all tasks (i.e., both the old and the new tasks). The results of Table 5 show that AdaHIT still achieves comparable performance but consumes less computational cost.

Method	ConvAI2	WoW	ED
MT (B-FT)	0.8878	0.9274	0.6241
MT (A-FT)	0.8767	0.9094	0.6136

Table 4: Results on the old tasks. MT (B-FT) and MT (A-FT) represent the multi-tasking model before and after being fine-tuned on the new task, respectively

ConvAI2	WoW	ED	BST	Average
0.8833	0.9233	0.6288	0.8342	0.8174

Table 5: Results of the model that is first pre-trained on the old tasks and then multi-tasked on all tasks.