PEK: A Parameter-Efficient Framework for Knowledge-Grounded Dialogue Generation

Pan Yang[1], **Dandan Song**[1]*, **Zhijing Wu**[1], **Yanru Zhou**[1], **Ziyi Yang**[2]
[1]School of Computer Science and Technology, Beijing Institute of Technology, China
[2]School of Cyberspace Science and Technology, Beijing Institute of Technology, China
{yangpan,sdd,zhijingwu,zhouyanru,yziyi}@bit.edu.cn

Abstract

Pre-trained language models (PLMs) have shown great dialogue generation capability in different scenarios. However, the huge VRAM consumption when fine-tuning them is one of their drawbacks. PEFT approaches can significantly reduce the number of trainable parameters, which enables us to fine-tune larger dialogue generation models. However, the reduction in parameter quantity can diminish a PLM's expressive capacity and affect the PLM's learning from certain specific examples like knowledge-related conversations. Previous works have demonstrated that injecting external knowledge into dialogue generation models can improve the model's performance in knowledge-related conversations. Nonetheless, these methods are designed for the scenario where most parameters of the entire framework are trainable. In this paper, we propose PEK, a parameter-efficient framework for knowledge-enhanced dialogue generation. It enables PLMs to leverage external knowledge documents and knowledge graphs to enhance its generation capabilities with an acceptable number of trainable parameters. Evaluation results on the Wizard of Wikipedia and CMU_DoG datasets show that our approach outperforms baseline methods on multiple evaluation metrics, which validates the effectiveness of our approach.

1 Introduction

The success of pre-trained models like BERT (Devlin et al., 2018), GPT-2 (Radford et al., 2019), BART (Lewis et al., 2019) and T5 (Raffel et al., 2020), demonstrates that more parameters and larger datasets, can lead to better language representation capabilities and great performances on various NLP tasks. Generally, fine-tuning on downstream task datasets helps further improve a pre-trained model's performance. However, since the advent of pre-trained language models, one of their drawbacks has been the huge VRAM consumption when fine-tuning on them. This issue is particularly prominent in large language models like GPT-3 (Brown et al., 2020) and LLaMA (Touvron et al., 2023). Millions and billions of parameters make it difficult to fine-tune them on a task-specific downstream dataset without a substantial number of high-performance GPUs and a significant amount of VRAM.

An effective solution is using parameter-efficient methods to reduce the number of trainable parameters. However, reducing the number of trainable parameters may lead to a decrease in model performance. Finding ways to maintain model performance is a challenging problem. Several approaches have been explored, like BitFit (Zaken et al., 2021), Adapter (Houlsby et al., 2019; Lin et al., 2020; Pfeiffer et al., 2021; Rücklé et al., 2021), P-tuning (Liu et al., 2021b, 2022), Prompt-tuning (Lester et al., 2021a), LoRA (Hu et al., 2021) and AdaLoRA (Zhang et al., 2023). Nonetheless, these methods still have some shortcomings, such as a significant drop in model performance on certain tasks, introduction of training or inference latency, or imprecise allocation of computational resources.

Using PEFT approaches, we can significantly reduce the number of trainable parameters, which enables us to fine-tune larger dialogue generation models. However, the reduction in parameter quantity can diminish a PLM's expressive capacity and affect the PLM's learning from certain specific examples like knowledge-related conversations. Pre-trained models acquire a considerable amount of knowledge during the pre-training process. In the scenario of full fine-tuning, the model can implicitly learn to apply this knowledge in downstream tasks during the fine-tuning process. Nevertheless, in the PEFT scenario, where the number of trainable parameters is significantly reduced, this transfer process may be affected. Moreover, the

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2024, pages 9261–9273
August 11-16, 2024 ©2024 Association for Computational Linguistics

knowledge obtained through pre-training is often not explicitly represented, which brings us challenges to explore better ways to utilize it. Some works (Li et al., 2019a; Kim et al., 2020; Zhao et al., 2020b; Zhan et al., 2021; Yang et al., 2022) have attempted to integrate external knowledge such as documents and knowledge graphs into dialogue generation models and achieved remarkable results. The achievements of these methods have demonstrated to us that introducing external knowledge helps to enhance a dialogue generation model's generative capabilities in knowledge-related conversations. Nonetheless, these methods are usually designed for the scenario where most parameters of the entire framework are trainable. How to inject external knowledge into PLMs to enhance its generation capabilities with a small number of trainable parameters in the entire framework remains a subject for further research.

To address the issues mentioned above, we present a parameter-efficient structure which can introduce external knowledge graph to PLMs. Our contributions in this paper can be summarized as: (1) We propose PEK, a **P**arameter-**E**fficient **K**nowledge-grounded dialogue generation framework, which enables PLMs to leverage external knowledge sources to generate knowledge-enriched responses and requires only a small number of trainable parameters; (2) We propose T-LoRA, which is based on SVD decomposition and can allocate more computational resources to weight matrices with less redundancy. T-LoRA does not require a dynamic parameter sensitivity estimation process during fine-tuning, which also means that it does not consume additional computational resources during training; (3) We propose a geometric mean weighted attention mechanism. It can effectively measure the importance of knowledge graph triplets and mitigate the issue of knowledge noise introduced by irrelevant triplets.

2 Methodology

2.1 Problem Statement

Knowledge-enhanced dialogue generation aims to develop a generative model that can leverage not only the dialogue history but also external knowledge sources like knowledge documents and knowledge graphs. Since pre-trained language models have better generation capabilities, fine-tuning them as the backbone network can further improve the model performance. However, a limited budget makes it difficult to fine-tune all the parameters. How to enhance the performance of the model when the number of trainable parameters is limited, is the problem to be solved.

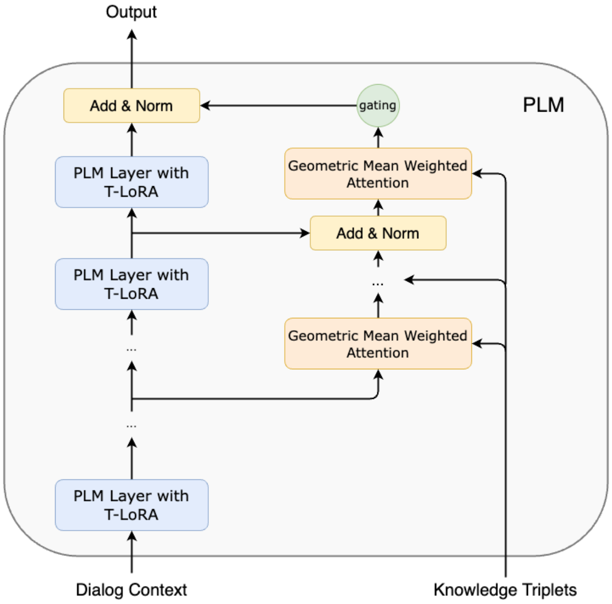


Figure 1: Overview of our method

Formally, we denote the dialogue history as $\mathcal{H}$, which encompasses all the historical tokens of the conversation, $\mathcal{H} = \{u_{Hi}\}$, where u_{Hi} is the i-th token of the dialogue history. The external knowledge sources are denoted as $\mathcal{K}$. In this paper, we mainly focus two commonly encountered types of external knowledge: knowledge documents and knowledge graphs. We denote the knowledge document as $\mathcal{K}_{\mathcal{D}} = \{u_{Di}\}$, where u_{Di} is the i-th token of the document and the knowledge graph as $\mathcal{K}_{\mathcal{G}} = \{(h_i, r_i, t_i)\}$, where h_i, r_i, t_i are respectively the head entity, relation and tail entity of the i-th tuple in the knowledge graph. The set of trainable parameters of the model is θ_t and the set of untrainable parameters is θ_f, where the size of θ_t is usually much smaller than θ_f. Our goal is to train a model with a PLM or LLM as its backbone to learn the conditional probability distribution $P(x_i|x_{1,2,...,j<i}; \mathcal{H}; [\mathcal{K}_{\mathcal{D}}; \mathcal{K}_{\mathcal{G}}]; [\theta_t; \theta_f])$.

2.2 Overview

Our method introduces T-LoRA into specific layers in PLMs. To enable the model to handle knowledge graphs, we insert knowledge fusion modules between certain layers. Fig. 1 presents an overview of our model.

2.3 T-LoRA

Before we start this section, let's take a glance at the vanilla LoRA. LoRA can be expressed as the following formula:

$$h = W_0 x + \frac{\alpha}{r} BAx, \tag{1}$$

where h is the output of a projection layer with an input x and $W_0 \in \mathbb{R}^{m\times n}$ is the original weight matrix of the PLM, which is not trainable during finetuning, $B \in \mathbb{R}^{m\times r}$ and $A \in \mathbb{R}^{r\times n}$ are learnable weight matrices. n, m are the numbers of input and output features. $r \ll \min(m, n)$, is a hyperparameter and α is the scaling factor. Aghajanyan et al. (2020) demonstrates that PLMs usually have a lower "intrinsic dimension", which may indicate the presence of a significant amount of redundant information within the weight matrices.

In the original implementation, all the matrices with LoRA have the same value for r. However, different weight matrices in the same PLM tend to have varying degrees of importance, which is demonstrated in Zhang et al. (2023)'s work. Thus selecting an appropriate r for each matrix may further improve parameter efficiency. Here, we perform singular value decomposition (SVD) on several weight matrices of DialoGPT$_{\text{large}}$(Zhang et al., 2019)[1] and define a function to measure the redundancy of matrix information:

$$f(W, \lambda) = \text{argmin}_k [\frac{\sum_{j=1}^{k} \sigma_j^2}{\sum_{j=1}^{N} \sigma_j^2} \geq \lambda], \tag{2}$$

where N is the number of the singular values of the matrix W and σ_j denotes the j-th singular value of W in descending order and $\lambda \in [0, 1]$ is a pre-determined parameter. Given a fixed λ, a low $f(W, \lambda)$ value means that W is more likely to contain redundant information[2]. Each line in Fig. 2 shows the variation trend of $f(W, \lambda)$ with respect to λ for certain matrices. It can be observed that there are differences in the redundancy of information among different matrices. Some matrices are highly information-redundant, which is further illustrated in Fig. 3.

In this paper, we propose a simple adaption to the vanilla LoRA, which is named as **T**hreshold **LoRA** (T-LoRA). To apply T-LoRA to a weight matrix W, we perform SVD to it first and denote the result as:

$$W = U\Sigma V^T. \tag{3}$$

Let the i-th singular values in Σ to be $\sigma_i (\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots)$, r is calculated with:

$$r = f(W, \lambda) = \text{argmin}_k [\frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{N} \sigma_i^2} \geq \lambda], \tag{4}$$

where N is the number of singular values of W and λ is a preselected threshold. T-LoRA can be represented by the following formula:

$$h = W_0 x + \alpha BAx, \tag{5}$$

. Unlike the vanilla LoRA, r is calculated with Eq. (4) and α is a manually-chosen scaling factor.

2.4 Knowledge Fusion

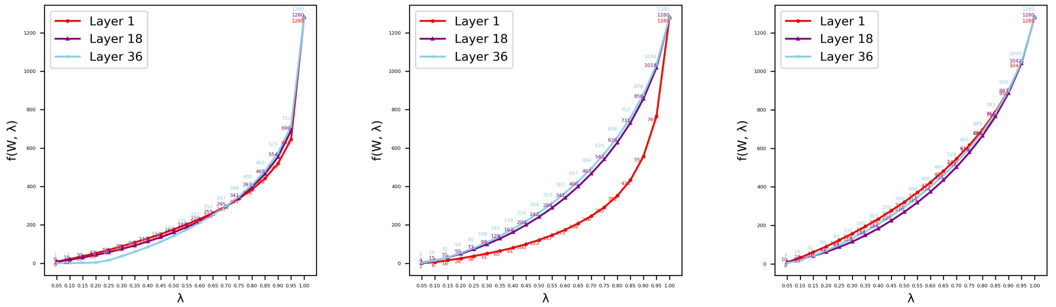
Our model takes knowledge documents and knowledge graphs as its external knowledge sources. For the knowledge document, we concatenate it with the dialogue history to obtain the input to the pre-trained model. The input token sequence is $\mathcal{I} = [\mathcal{K}_\mathcal{D}; \mathcal{H}] = [u_{D1}, u_{D2}, \dots, [SEP], u_{H1}, u_{H2}, \dots]$.

For knowledge triplets, we employ a different approach. Firstly, we split the set of triplets into three sequences: the head entity sequence, the relation sequence, and the tail entity sequence. We denote the matrix composed of embedding vectors of head entities, relations and tail entities as H, R, T, where the i-th row of H, R, T is respectively the embedding vector of the head entity, relation and tail entity of the i-th triplet. We employ DistMult (Yang et al., 2015) to pretrain the entity embedding matrix E_e and relation embedding matrix E_r. Then we freeze all the parameters of E_e and E_r.

Instead of directly employing vanilla multi-head attention (Vaswani et al., 2017), we use a graph attention mechanism which takes the similarity between tokens and the head entities, relations and tail entities into consideration, to facilitate information interaction between the tokens and the knowledge graph. We call it Geometric Mean Weighted Attention. We select a few specific layers to insert the geometric mean weighted attention modules. The set of indices of the selected layers is denoted as $\mathcal{S}$ and s_i is the i-th element of $\mathcal{S}$ in ascending

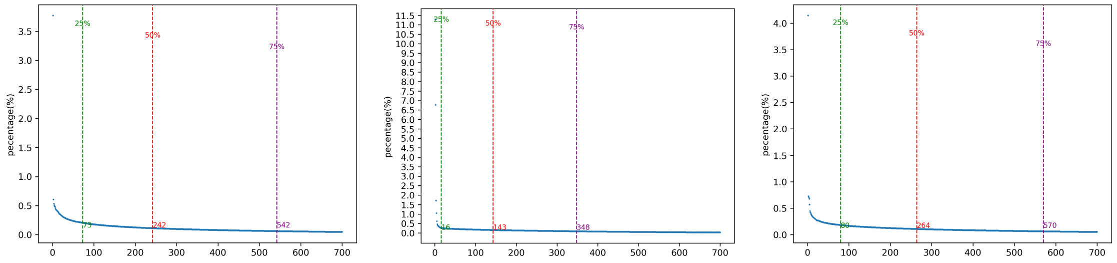
[1] https://huggingface.co/microsoft/DialoGPT-large

[2] Let's consider an extreme example, where $A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$ and $B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$. $f(A, 1.0) = 1, f(B, 1.0) = 2$. It can be observed that A contains redundant information since the second row can be linearly represented by the first row.



(a) Self-Attention Out-Projection(W_O) (b) MLP Up-Projection (c) MLP Down-Projection

Figure 2: $f(W, \lambda)$ for different matrices of different layers and λ values.



(a) MLP Up-Projection (Layer 18) (b) Self-Attention Out-Projection(W_O) (Layer 36) (c) MLP Up-Projection (Layer 36)

Figure 3: $\frac{\sigma_i^2}{\sum_{i=1}^{N} \sigma_i^2}$ for different matrices

order. Let $H_O^{s_i}$ be the output of the s_i-th layer and $H_{O_{\mathcal{K}}}^{s_{i-1}}$ be the output of the graph attention module which is inserted after the s_{i-1}-th layer. We project $H_O^{s_i} + H_{O_{\mathcal{K}}}^{s_{i-1}}$ into a subspace of dimension k, where k is the dimension of the entity and relation embedding matrix. The projection result is denoted as $H_P^{s_i}$. Then, we separately compute the attention scores for the i-th head entity ($W_H^{s_i}$), relation ($W_R^{s_i}$), and tail entity($W_T^{s_i}$):

$$W_H^{s_i} = \text{softmax}(\frac{(W_{Q_H} H_P^{s_i})(W_H H)^T}{\sqrt{d_k}}) \quad (6)$$

$$W_R^{s_i} = \text{softmax}(\frac{(W_{Q_R} H_P^{s_i})(W_R R)^T}{\sqrt{d_k}}) \quad (7)$$

$$W_T^{s_i} = \text{softmax}(\frac{(W_{Q_P} H_P^{s_i})(W_T T)^T}{\sqrt{d_k}}), \quad (8)$$

where d_k is the dimension of each attention head and $W_{Q_H}, W_{Q_R}, W_{Q_P}, W_H, W_R, W_T$ are projection matrices. The final attention weight matrix W^{s_i} is:

$$W^{s_i} = \text{softmax}(\sqrt[3]{W_H^{s_i} \odot W_R^{s_i} \odot W_T^{s_i}}), \quad (9)$$

where $A \odot B$ is the Hadamard product of A, B. Fig. 4 shows an overview of the calculation process.

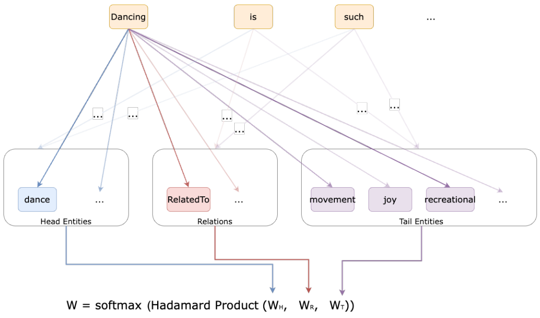


Figure 4: An overview of the calculation process. The example used in this figure is the same as Tab. 4. A more opaque line represents a larger attention weight.

Then we weight the information of each tail entity according to the weight matrix W^{s_i}. The results are fed into a fully connected network as the final output:

$$H_{O_{\mathcal{K}}}^{s_i} = FFN(W^{s_i}(W_{V_T} T)) \quad (10)$$

(W_{V_T} is the projection matrix). In addition, we use a knowledge gating mechanism to mitigate the noise introduced by external knowledge, which can be described as:

$$p = \text{Sigmoid}(W^{f}_{O_{\mathcal{K}}} H^{f}_{O_{\mathcal{K}}}) \tag{11}$$

$$H_O = pH^{f}_{O_{\mathcal{K}}} + (1-p)H^{f}_{O}, \tag{12}$$

where $H^{f}_{O_{\mathcal{K}}}$ is the output of the last graph attention module, H^{f}_{O} is the output of the PLM and H_O is the final output of the stacked transformer layers.

3 Experiment Settings

3.1 Datasets

We conduct experiments on two public datasets: Wizard of Wikipedia (Dinan et al., 2019) and CMU_DoG (Zhou et al., 2018).

Wizard of Wikipedia Wizard of Wikipedia (WoW) is a document-grounded dialogue dataset. The dataset primarily consists of dialogues between two agents. One of the agents who is called "the wizard", plays the role of an expert. The other is the apprentice. The wizard has access to the knowledge documents retrieved from Wikipedia while the apprentice does not. The apprentice asks questions and the wizard provides responses with retrieved documents. The Wizard of Wikipedia dataset covers approximately 1.3K topics.

CMU_DoG CMU_DoG is another dialogue dataset based on knowledge documents, which contains conversations between two individuals centered around a particular movie. Unlike WoW, both speakers in CMU_DoG have access to the knowledge documents, but the content they can access may be different. More details about the datasets can be found in Appendix A.

Knowledge Base For the WoW dataset, we use ConceptNet (Speer et al., 2017)[3] , which is an open, multilingual knowledge graph containing commonsense knowledge, as our knowledge base. ConceptNet includes a great number of concepts and the connections between them. ConceptNet is designed to help computers understand various concepts that people use in everyday life. It is an important knowledge source for dialogue systems and other NLP applications. In our experiments, we utilize the subgraph of ConceptNet where the source language is English. We use spaCy [4] to recognize named entities. For the WoW dataset, we extract 39,101 triplets, involving 26,684 entities and 17 types of relationships.

Since the CMU_DoG dataset is highly movie-centric, we employ YAGO (Suchanek et al., 2023), a knowledge graph that incorporates more movie-related knowledge, as our knowledge base. We collect a total of 74,350 entities, 10 types of relations, and 82,861 triplets for the CMU_DoG dataset.

3.2 Metrics

We mainly use automatic metrics to evaluate the performance of our method: (1) Overlapping-based metrics: **BLEU-4** (Papineni et al., 2002), **ROUGE-L** (Lin, 2004); (2) Embedding-based metrics (Liu et al., 2016): Greedy Matching (**GM**), Embedding Average (**EA**) and Vector Extrema (**VE**). All the automatic evaluation results are computed with the nlg-eval toolkit (Sharma et al., 2017) [5].

3.3 Baselines

To validate the effectiveness of our method, we compare it with other state-of-the-art parameter-efficient methods. We select full fine-tuning as a strong baseline. We compare our method with the following methods: (1) **Adapter tuning**: **Adapter-H** (Houlsby et al., 2019), **Adapter-L** (Lin et al., 2020); (2) **BitFit** (Zaken et al., 2021); (3) **P-Tuning** (Liu et al., 2021b) and **P-Tuning V2** (Liu et al., 2022); (4) **PrefixTuning** (Li and Liang, 2021); (5) **LoRA** (Hu et al., 2021) and **AdaLoRA** (Zhang et al., 2023). For PEFT modules other than Adapter and BitFit, we utilize the implementation provided by Hugging Face [6].

In addition, we also compare our method with the following high-performing knowledge-enhanced models: **(1) Transformer Memory Network (TMN)**, which is the model proposed along with the WoW Dataset in (Dinan et al., 2019). **(2) Incremental Transformer with Deliberation Decoder (ITDD)** (Li et al., 2019b), which is a transformer-based model and can encode multi-turn dialogue history and knowledge incrementally and generate responses with a deliberation decoder. **(3) Disentangled Response Decoder (DRD)** (Zhao et al., 2020a), which is a model for low-resource scenario and tackles this challenge by using pre-training techniques. **(4) KnowledGPT** (Zhao et al., 2020b) , which consists of a pre-trained language model for response generation

[3] https://conceptnet.io/

[4] https://github.com/explosion/spaCy

[5] https://github.com/Maluuba/nlg-eval

[6] https://github.com/huggingface/peft

and a knowledge selection module. Both models are jointly optimized with an unsupervised method. **(5) CoLV** (Zhan et al., 2021), a collaborative latent variable model that can integrate knowledge selection and knowledge-aware response generation simultaneously in separate yet collaborative latent spaces. **(6) TAKE** (Yang et al., 2022), which annotates the topic shift and topic inheritance labels in multi-round dialogues with distant supervision and alleviate the noise problem in pseudo labels through curriculum learning and knowledge distillation.

4 Experimental Results and Discussions

4.1 Results and Analysis

The implementation details can be found in Appendix B. Table 1 and Table 2 present the automatic evaluation results on Wizard of Wikipedia and CMU_DoG. Based on the experimental results presented in the two tables, we have the following observations: (1) On most automatic metrics, our approach outperforms the PEFT baselines and even some of the non-PEFT models, which indicates the effectiveness of our approach in knowledge-enhanced dialogue generation. (2) The performances of *T-LoRA($\lambda = 0.1$) + kg*, *LoRA($r = 20$) + kg* and *AdaLoRA($r = 20$) + kg* are even better than full fine-tuning. This improvement may be attributed to the denoising effect of the low-rank approximation of the increment matrix, as also mentioned in (Hu et al., 2021). (3) AdaLoRA and our method exhibit similar performances on automatic metrics. However, compared to AdaLoRA, our method does not require dynamic evaluation of the sensitivity of each parameter during training. It is much simpler and does not introduce an additional computational process during training since all the r values can be pre-computed.

Table 3 shows the human evaluation results. It can be observed that while *KnowledGPT*, *AdaLoRA($r = 20$) + kg* and *T-LoRA($\lambda = 0.1$) + kg* are comparable in terms of language fluency, our method performs better on context coherence and knowledge relevance, which is consistent with the results on automated evaluation metrics. All the Kappa values are not less than 0.6, indicating a relatively consistent agreement among human experts. More details about the human evaluation can be found in Appendix C.

4.2 Ablation Study

To investigate the impact of geometric mean weighted graph attention on performance, we compare our method with the following variants: (1) *T-LoRA w/o kg*: The geometric mean weighted graph attention modules are removed; (2) *T-LoRA + A-Attn*: The final attention weight $W^{s_i} = \frac{W_H^{s_i} + W_R^{s_i} + W_T^{s_i}}{3}$, is the arithmetic mean of $W_H^{s_i}, W_R^{s_i}, W_T^{s_i}$, rather than the geometric mean; (3) *T-LoRA + V-Attn*: The geometric mean weighted graph attention modules are replaced with vanilla multihead attention modules. The evaluation results are reported in Table 1 and Table 2.

We observe that (1) removing the geometric mean weighted graph attention modules leads to a performance drop on both the Wizard of Wikipedia and CMU_DoG datasets, which confirms the effectiveness of the geometric mean weighted graph attention modules; (2) *T-LoRA + A-Attn* and *T-LoRA + V-Attn* also exhibit some performance degradation, which may be attributed to the fact that, in certain cases, they struggle to effectively measure the relevance between knowledge triplets and the context of the dialogue. Additionally, we conduct a human evaluation for *T-LoRA w/o kg*, and the results are presented in Table 3. *T-LoRA w/o kg* scores lower in terms of contextual coherence and knowledge relevance compared to the full model, validating the effectiveness of the geometric mean weighted graph attention mechanism.

4.3 Case Study

Table 4 presents an example from the Test Seen dataset of Wizard of Wikipedia. From this example, it can be observed that our model can leverage the knowledge triplets correctly and the generated response is more coherent with the context and closer to the human-written reply, while KnowledGPT exhibits issues with repetitive generation and the response generated by TMN is not coherent enough with the context.

5 Related Work

5.1 Parameter-Efficient Fine-Tuning

The massive number of parameters makes it challenging to fine-tune all the parameters of a pre-trained model. To address this issue, parameter-efficient fine-tuning (PEFT) is proposed.

PEFT aims to find a method to significantly reduce the number of trainable parameters required

Method	#Trainable Parameters / # Parameters (%)	Seen					Unseen				
		BLEU-4	ROUGE-L	EA	VE	GM	BLEU-4	ROUGE-L	GM	VE	EM
TMN	-	1.7	13.7	0.844	0.427	0.658	0.9	11.3	0.839	0.408	0.645
ITDD	-	2.5	-	0.841	0.425	0.654	1.1	-	0.826	0.364	0.624
DRD	-	5.5†	-	0.835 †	0.434 †	0.658†	4.3†	-	0.828†	0.422†	0.628†
KnowledGPT	> 227M	5.8	17.8	0.872	0.463	0.685	4.7	16.6	0.870	0.452	0.674
CoLV	-	2.9	-	-	-	-	2.1	-	-	-	-
TAKE	> 227M	3.6	20.5	-	-	-	3.3	18.3	-	-	-
Full Fine-Tuning + kg	775.5M (100%)	5.8	18.5	0.856	0.485	0.684	5.4	17.9	0.855	0.473	0.676
BitFit + kg	2.0M (0.25%)	3.5	15.0	0.838	0.464	0.667	3.2	14.6	0.838	0.454	0.660
P-Tuning + kg	6.5M (0.82%)	5.0	17.1	0.836	0.466	0.664	4.6	16.5	0.834	0.459	0.659
LoRA($r = 8$) + kg	7.4M (0.94%)	5.5	18.1	0.855	0.482	0.683	5.1	17.5	0.852	0.472	0.675
AdaLoRA($r = 8$) + kg	7.4M (0.94%)	5.5	18.3	0.856	0.483	0.685	5.1	17.7	0.854	0.474	0.678
Adapter-H + kg	7.5M (0.95%)	5.4	18.0	0.853	0.480	0.680	4.9	17.4	0.849	0.469	0.672
Adapter-L + kg	7.5M (0.95%)	5.4	17.9	0.851	0.478	0.676	4.8	17.4	0.849	0.467	0.671
T-LoRA($\lambda = 0.05$) + kg	6.3M (0.81%)	**5.6**	**18.4**	**0.857**	**0.485**	**0.687**	**5.2**	**17.9**	**0.856**	**0.476**	**0.679**
PrefixTuning + kg	15.0M (1.89%)	5.3	17.6	0.848	0.477	0.677	4.7	17.0	0.847	0.468	0.670
P-TuningV2 + kg	15.0M (1.89%)	5.3	17.8	0.849	0.479	0.679	4.8	17.0	0.845	0.467	0.670
Adapter-H + kg	15.6M (1.97%)	5.5	18.1	0.852	0.482	0.681	4.9	17.5	0.850	0.472	0.673
Adapter-L + kg	15.6M (1.97%)	5.4	18.1	0.851	0.483	0.684	4.9	17.4	0.852	0.473	0.674
AdaLoRA($r = 20$) + kg	16.2M (2.04%)	5.9	18.4	0.856	0.486	0.685	5.2	17.9	0.853	0.475	0.676
LoRA($r = 20$) + kg	16.2M (2.04%)	5.8	18.6	0.855	0.486	0.685	5.4	18.0	0.853	0.474	0.676
T-LoRA($\lambda = 0.1$) w/o kg	14.4M (1.83%)	5.7	18.5	0.850	0.483	0.683	5.1	17.6	0.852	0.471	0.668
T-LoRA($\lambda = 0, 1$) + V-Attn	14.9M (1.88%)	5.8	18.5	0.856	0.485	0.684	5.3	17.9	0.855	0.477	0.677
T-LoRA($\lambda = 0.1$) +kg	15.9M (2.00%)	**6.1**	**18.8**	**0.857**	**0.488**	**0.686**	**5.4**	**18.1**	**0.855**	**0.476**	**0.678**
T-LoRA($\lambda = 0.1$) + A-Attn	15.9M (2.00%)	5.9	18.6	0.856	0.486	0.685	5.2	17.9	0.854	0.475	0.676

Table 1: Evaluation Results (mean of 3 runs) on Wizard of Wikipedia. Numbers in **Bold** fonts indicate the improvement to the baseline methods with similar parameter counts is statistically significant (t-test with p-value $<$ 0.05). Numbers marked with "†" are the results reported in (Zhao et al., 2020a). We adjust hyperparameters to make the number of trainable parameters comparable across different PEFT methods. Numbers with a red square box are the best results among all the baselines.

Method	BLEU-4	ROUGE-L	GM	VE	EM
TMN	0.6	-	0.802	0.351	0.617
ITDD	0.9	-	0.748	0.390	0.587
DRD	1.2†	-	0.809†	0.413†	0.633†
KnowledGPT	-	-	0.837†	0.437†	0.654†
CoLV	0.6	-	-	-	-
TAKE	0.7	10.2	-	-	-
Full Fine-Tuning + kg	0.9	9.9	0.792	0.427	0.623
BitFit + kg	0.5	8.8	0.739	0.438	0.614
P-Tuning + kg	0.6	9.6	0.756	0.424	0.611
LoRA($r = 8$) + kg	0.8	10.0	0.763	0.441	0.623
AdaLoRA($r = 8$) + kg	0.8	10.2	0.766	0.443	0.625
Adapter-H + kg	0.8	10.1	0.760	0.440	0.620
Adapter-L + kg	0.7	10.0	0.762	0.438	0.623
T-LoRA($\lambda = 0.05$) + kg	0.8	**10.3**	**0.769**	**0.444**	**0.627**
PrefixTuning + kg	0.7	9.5	0.759	0.425	0.616
P-TuningV2 + kg	0.7	9.6	0.757	0.428	0.614
Adapter-L + kg	0.8	9.8	0.760	0.433	0.621
Adapter-H + kg	0.8	9.9	0.763	0.436	0.622
AdaLoRA($r = 20$) + kg	0.8	10.0	0.769	0,444	0.627
LoRA($r = 20$) + kg	0.8	10.0	0.767	0.442	0.626
T-LoRA($\lambda = 0.1$) w/o kg	0.8	9.8	0.769	0.422	0.622
T-LoRA($\lambda = 0, 1$) + V-Attn	0.9	10.4	0.768	0.440	0.629
T-LoRA($\lambda = 0.1$) + kg	1.0	**10.7**	**0.774**	**0.445**	**0.630**
T-LoRA($\lambda = 0.1$) + A-Attn	1.0	10.5	0.772	0.443	0.629

Table 2: Evaluation Results (mean of 3 runs) on the test set of CMU_DoG. Numbers marked with "†" are the results reported in (Zhao et al., 2020a) and (Zhao et al., 2020b).

for fine-tuning a PLM. Adapter-tuning (Houlsby et al., 2019) inserts additional adapter layers after the attention and feed-forward module of each transformer layer. Prompt Tuning (Lester et al., 2021b) assumes a prefix prompt with a fixed length for each input text. A soft prompt is parameterized by neural networks and updated during fine-tuning on downstream tasks while the parameters of the PLM remain frozen. Prefix Tuning, as proposed in Li and Liang (2021)'s work, add a prefix which is composed of a sequence of virtual tokens to the input. During training, only the embedding matrix of virtual tokens remains trainable. Compared with Prefix-Tuning, P-Tuning (Liu et al., 2021b) introduces differentiable virtual tokens as well but only inserts them into the input layer instead of each layer. Moreover, the insertion position is selectable. The introduced virtual tokens are encoded by a prompt encoder, not randomly initialized. Ben Zaken et al. (2022) propose a sparse-finetuning method where only the bias tensors of a PLM are kept trainable.

LoRA, introduced in (Hu et al., 2021), is based on the observation that PLMs typically process a lower intrinsic rank. This approach treats the fine-tuned weight matrix as the sum of the original weight matrix and an increment matrix. It decomposes the increment matrix into the product of two trainable matrices with an extremely low rank. In this way, the number of trainable parameters can be reduced significantly. AdaLoRA, proposed by Zhang et al. (2023), combines LoRA and parameter pruning together. Instead of setting the hyperparameter r of each matrix to the same value, it estimates the importance and sensibility of each module and allocates more trainable parameters to

Method	**Test Seen**				**Test UnSeen**				**CMU_DoG**			
	Fluency	Context Coherence	Knowledge Relevance	Kappa	Fluency	Context Coherence	Knowledge Relevance	Kappa	Fluency	Context Coherence	Knowledge Relevance	Kappa
KnowledGPT	1.76	1.64	1.67	0.68	1.71	1.60	1.65	0.67	-	-	-	-
AdaLoRA($r = 20$) + kg	1.76	1.72	1.81	0.71	1.74	1.62	1.76	0.73	1.64	1.53	1.60	0.76
T-LoRA($\lambda = 0.1$) + kg	1.79	1.74	1.86	0.74	1.79	1.69	1.79	0.64	1.66	1.57	1.69	0.70
T-LoRA($\lambda = 0.1$) w/o kg	1.71	1.57	1.60	0.65	1.74	1.55	1.52	0.66	1.60	1.50	1.43	0.77

Table 3: Human Evaluation Results on Wizard of Wikipedia and CMU_DoG.

Dialogue Context	**A:** Dancing is such a fun activity, though I'm not very good at it. Are you?
Retrieved Document	Lion dance is a form of traditional dance in Chinese culture and other Asian countries in which performers mimic a lion's movements in a lion costume to bring good luck and fortune ...
Relevant Triplets	(dance, RelatedTo, movement), (dance, RelatedTo, joy) (dance, RelatedTo, recreational), (dance, RelatedTo, people) ...
TMN	I do not dance, but I do enjoy performing art.
KnowledGPT	I am not very good at dancing. I am not good at dancing. I am not good at dancing.
T-LoRA($\lambda = 0.1$) + kg (Ours)	I am not very good at it either, but I do enjoy it. It is one of the most popular recreational activities in the world.
Human Written	I do like to dance! Dancing has symbolic cultural meaning across the world.

Table 4: An example sampled from the Test Seen dataset of Wizard of Wikipedia

matrices that appear more significant. The experimental results demonstrate that a more accurate allocation of computational resources can further improve the performance of LoRA.

5.2 Knowledge-Enhanced Dialogue Generation

While traditional dialogue generation models perform well on several generation tasks, they still struggle with issues such as out-of-context responses and illusions. The approaches of external knowledge enhancement for dialogue generation aim to address these issues. The key of these models is to incorporate rich external knowledge sources to enhance the response generation capability of a dialogue generation model.

Previous works investigate methods for incorporating external knowledge in various forms into dialogue generation models. Dinan et al. (2019); Li et al. (2019b); Kim et al. (2020) explore techniques for incorporating unstructured documents into dialogue generation models through document retrieval and knowledge selection. Moon et al. (2019); Tuan et al. (2019) focus on knowledge injection with external knowledge graphs. Mostafazadeh et al. (2017); Huber et al. (2018) investigate how to incorporate visual information into dialogue generation. Since it may be difficult to obtain enough training samples with accurate knowledge annotations in a new domain, there are also models designed for knowledge-grounded dialogue generation in low-resource scenarios: Zhao et al. (2020a); Liu et al. (2021a).

Pre-trained language models, like GPT-2 (Radford et al., 2019), BART (Lewis et al., 2019) and T5 (Raffel et al., 2020), have demonstrated excellent performance in various generative tasks. There is also work dedicated to integrating external knowledge with such pre-trained models. Zhao et al. (2020b) leverages BERT (Devlin et al., 2018) to construct a knowledge selection module to choose relevant knowledge documents for dialogue generation, assisting the GPT-2 model in generating appropriate responses. The authors employ an unsupervised approach to jointly optimize two pre-trained models, achieving impressive performances on the test datasets. Zhan et al. (2021) propose a collaborative latent variable (CoLV) to integrate knowledge selection and response generation simultaneously and capture the the inherent correlation between them. Yang et al. (2022) propose a topic-shift aware knowledge selector. It annotates topic shift and topic inheritance labels in multi-turn dialogues with distant supervision, and mitigates the noise issue in pseudo labels through curriculum learning and knowledge distillation.

6 Conclusion

In this paper, we propose PEK, a parameter-efficient framework for knowledge-grounded dialogue generation. We make an improvement to the vanilla LoRA, which can allocate more computational resources to matrices with less information redundancy. Moreover, to incorporate knowledge graphs into dialogue generation, we introduce a geometric mean weighted mechanism. The evaluation results on both the Wizard of Wikipedia dataset and the CMU_DoG dataset have shown that our method outperforms all the PEFT baselines and some of the non-PEFT methods.

Limitations

Although our approach achieve excellent performances on both datasets, it still has some limitations: (1) Our method relies on the assumption that there is not a significant change in the redundancy of the weight matrix information of the pre-trained language model before and after fine-tuning, which may not hold true for some specific tasks; (2) The DistMult method we used in our experiments is more adept at handling symmetric patterns in knowledge graphs, while in typical knowledge graphs, there are much more asymmetric patterns, which may lead to a performance degradation.

Ethics Statement

Our work relies primarily on publicly available datasets. We adhere to the policies regarding the use of this data and ensure that it does not raise copyright-related issues. In addition, our model has some limitations. In certain situations, its outputs may be unpredictable. It may generate inaccurate or biased responses. Therefore, we recommend conducting safety checks on model outputs if applied to human interactive systems.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (Grant No. 2022YFC3302100).

References

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. 2020. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *CoRR*, abs/2012.13255.

Jiaqi Bai, Zhao Yan, Ze Yang, Jian Yang, Xinnian Liang, Hongcheng Guo, and Zhoujun Li. 2023. Knowprefix-tuning: A two-stage prefix-tuning framework for knowledge-grounded dialogue generation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 525–542. Springer.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

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*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Bernd Huber, Daniel McDuff, Chris Brockett, Michel Galley, and Bill Dolan. 2018. Emotional dialogue generation using image-grounded language models. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA. Association for Computing Machinery.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential Latent Knowledge Selection for Knowledge-Grounded Dialogue. In *ICLR*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021a. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021b. The power of scale for parameter-efficient prompt tuning. *CoRR*, abs/2104.08691.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019a. Incremental transformer with deliberation decoder for document grounded conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 12–21, Florence, Italy. Association for Computational Linguistics.

Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019b. Incremental transformer with deliberation decoder for document grounded conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 12–21, Florence, Italy. Association for Computational Linguistics.

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, and Pascale Fung. 2020. Exploring versatile generative language model via parameter-efficient transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 441–459, Online. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Shilei Liu, Xiaofeng Zhao, Bochao Li, Feiliang Ren, Longhui Zhang, and Shujuan Yin. 2021a. A Three-Stage Learning Framework for Low-Resource Knowledge-Grounded Dialogue Generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2262–2272, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv:2103.10385*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854, Florence, Italy. Association for Computational Linguistics.

Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios Spithourakis, and Lucy Vanderwende. 2017. Image-grounded conversations: Multimodal context for natural question and response generation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 462–472, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Shrimai Prabhumoye, Kazuma Hashimoto, Yingbo Zhou, Alan W Black, and Ruslan Salakhutdinov. 2021. Focused attention improves document-grounded generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4274–4287, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. 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. *Journal of Machine Learning Research*, 21(140):1–67.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. AdapterDrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Fabian Suchanek, Mehwish Alam, Thomas Bonald, Pierre-Henri Paris, and Jules Soria. 2023. Integrating the wikidata taxonomy into yago.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yi-Lin Tuan, Yun-Nung Chen, and Hung-yi Lee. 2019. DyKgChat: Benchmarking dialogue generation grounding on dynamic knowledge graphs. 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 1855–1865, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yan Xu, Etsuko Ishii, Samuel Cahyawijaya, Zihan Liu, Genta Indra Winata, Andrea Madotto, Dan Su, and Pascale Fung. 2022. Retrieval-free knowledge-grounded dialogue response generation with adapters. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 93–107, Dublin, Ireland. Association for Computational Linguistics.

Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases.

Chenxu Yang, Zheng Lin, Jiangnan Li, Fandong Meng, Weiping Wang, Lanrui Wang, and Jie Zhou. 2022. TAKE: Topic-shift aware knowledge sElection for dialogue generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 253–265, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *CoRR*, abs/2106.10199.

Haolan Zhan, Lei Shen, Hongshen Chen, and Hainan Zhang. 2021. CoLV: A collaborative latent variable model for knowledge-grounded dialogue generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2250–2261, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *CoRR*, abs/1911.00536.

Xueliang Zhao, Wei Wu, Chongyang Tao, Can Xu, Dongyan Zhao, and Rui Yan. 2020a. Low-resource knowledge-grounded dialogue generation.

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020b. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390, Online. Association for Computational Linguistics.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 708–713, Brussels, Belgium. Association for Computational Linguistics.

Furthermore, Tables 6 presents the results of our method compared to some other baselines.

Method	#Trainable Parameters / # Parameters (%)	Seen					Unseen					CMU_DoG				
		BLEU-4	ROUGE-L	EA	VE	GM	BLEU-4	ROUGE-L	GM	VE	EM	BLEU-4	ROUGE-L	GM	VE	EM
LoRA ($r = 20$) w/o kg	14.7M	5.7	18.3	0.853	0.484	0.684	5.1	17.7	0.851	0.473	0.674	0.8	9.9	0.767	0.421	0.623
AdaLoRA ($r = 20$) w/o kg	14.7M	5.7	18.4	0.855	0.482	0.683	5.2	17.8	0.854	0.474	0.672	0.7	9.6	0.769	0.422	0.624

Table 5: Evaluation Results on Wizard of Wikipedia and CMU_DoG

Method	Seen			Unseen			CMU_DoG		
	F1	BLEU-4	ROUGE-L	F1	BLEU-4	ROUGE-L	F1	BLEU-4	ROUGE-L
KnowExpert(w) (Xu et al., 2022)	18.7	-	-	16.7	-	-	12.5	-	-
GPT2 + KnowPrefix-Tuning (Bai et al., 2023)	20.1	-	-	18.0	-	-	14.1	-	-
BART + KnowPrefix-Tuning (Bai et al., 2023)	20.3	-	-	18.3	-	-	14.6	-	-
Fid-RAG DPR-Poly(BART) (Shuster et al., 2021)	22.1	4.1	-	22.1	3.8	-	15.2	0.5	-
DoHa (Prabhumoye et al., 2021)	31.8	8.2	21.8	29.0	6.6	19.6	22.8	20.9	20.4
T-LoRA($\lambda = 0.1$) + kg	23.9	6.1	18.8	22.6	5.4	18.1	14.1	1.0	10.7

Table 6: Evaluation Results of more baselines on Wizard of Wikipedia and CMU_DoG

A Statistics of the datasets

Table 7 and Table 8 show more detailed statistics of the datasets.

Dataset	#Utterances	#Dialogues	#Topics	Avg.# of turns / dialogue
Train	166,787	18,430	1,247	9.0
Valid	17,715	1,948	599	9.1
Test Seen	8,715	965	533	9.0
Test Unseen	8,782	968	58	9.1

Table 7: Statistics of WoW

Dataset	#Utterances	#Dialogues	#Topics	Avg.# of turns / dialogue
Train	74,717	3,373	30	22.2
Valid	4,993	229	30	21.8
Test	13,646	619	30	22.0

Table 8: Statistics of CMU_DoG

B Implementation Details

We choose DialoGPT$_{\text{large}}$(Zhang et al., 2019), a model with 774M parameters and stronger dialogue generation capabilities compared to the original GPT-2 model, as our base model. We apply T-LoRA to all the weight matrices in the base model. To avoid introducing knowledge noise when the base model is capturing basic information such as phrases and grammar, we insert the geometric mean weighted graph attention modules into the last two layers of the base model. In our experiments, We employ the AdamW optimizer (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. For both WoW and CMU_DoG, the batch size is set to 6 and the scaling factor α is set to 2. The learning rate is set to 0.0001 initially and decreases linearly during the training process. We conduct experiments using an NVIDIA A100 GPU with 40GB of VRAM and train our model for 12 epochs. The checkpoint with the best performances on the validation set when the loss on the validation set no longer decreases is used to evaluate on the test set.

For TMN, we use the code released by the authors at https://github.com/facebookresearch/ParlAI/tree/main/projects/wizard_of_wikipedia. For ITDD, we utilize the code released by the authors at https://github.com/lizekang/ITDD. For KnowledGPT, we choose the implementation shared at https://github.com/zhaoxlpku/KnowledGPT.

C Details about Human Evaluation

We randomly collect 140 samples from Test Seen, Test Unseen and the test set of CMU_DoG respectively for human evaluation. Each sample consists of a dialogue history, relevant documents and triplets extracted from the knowledge base, and the response generated by the model with beam size = 3. We invite 3 graduates who are fluent in English to evaluate the generated responses from 3 different perspectives: *Fluency*, *Context Coherence* and *Knowledge Relevance*. The scores are assigned from $\{0, 1, 2\}$, where 0 represents poor and 2 represents good. We utilize Kappa (Fleiss, 1971) to measure the agreement among the human anotators. Due to the time-consuming nature of manual evaluation, we only select a few strong baseline models.

Operation	Time Complexity
SVD decomposition	$O(n^3)$
Calculation of r (each matrix)	$O(d)$
Forward Propagation (each layer)	$O(L^2d + rd^2L)$
Calculation of Graph Attention Weights (each example)	$O(LT)$

Table 9: Time Complexity of Different Computational Processes

D Additional Evaluation Results

Table 5 presents some evaluation results that we have not reported in the main body due to space limitations.

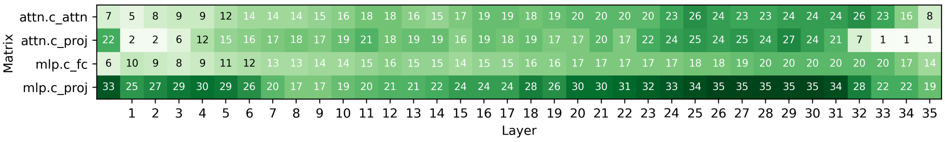


Figure 5: The r value of each incremental matrix ΔW for $\lambda = 0.1$

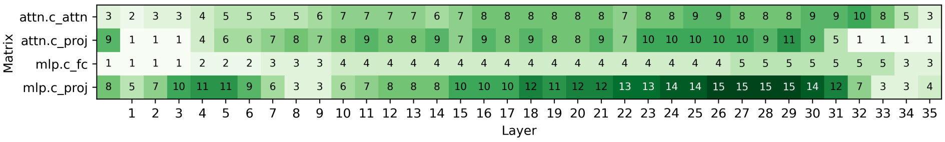


Figure 6: The r value of each incremental matrix ΔW for $\lambda = 0.05$

Scientific Aritifact	Lincense
GPT-2$_{\text{large}}$	MIT Lincense
DialoGPT$_{\text{large}}$	MIT Lincense
Wizard of Wikipedia	MIT Lincense
ConceptNet	Creative Commons Attribution-ShareAlike 4.0 International License
YAGO	Creative Commons Attribution 4.0 International License

Table 10: Lincenses of the scientific aritifacts used in this paper

The baseline results are reported by the authors in the respective papers.

E Time Efficiency Analysis

Table 9 shows the time complexity of different computational processes. Here, d is the hidden size of the PLM. L is the length of the token sequence and T is the number of relevant knowledge triplets.

In our experimental setup, the training speed is approximately 0.4 seconds per batch, and the inference speed is about 4.4 seconds per batch.

F Rank Distribution

Fig. 5 and Fig. 6 show the r value of each incremental matrix in DialoGPT$_{\text{large}}$ for $\lambda = 0.1$ and $\lambda = 0.05$ respectively.

G Lincenses

Table 10 shows the lincenses the scientific aritifacts used in our paper.