Towards Better Parameter-Efficient Fine-Tuning for Large Language Models: A Position Paper

Chengyu Wang¹, Junbing Yan², Wei Zhang², Jun Huang¹

¹ Alibaba Group, Hangzhou, China

² East China Normal University, Shanghai, China
{chengyu.wcy, huangjun.hj}@alibaba-inc.com,
{junbingyan531, zhangwei.thu2011}@gmail.com

Abstract

This paper delves into the pressing need in Parameter-Efficient Fine-Tuning (PEFT) for Large Language Models (LLMs). While LLMs possess remarkable capabilities, their extensive parameter requirements and associated computational demands hinder their practicality and scalability for real-world applications. Our position paper highlights current states and the necessity of further studying into the topic, and recognizes significant challenges and open issues that must be addressed to fully harness the powerful abilities of LLMs. These challenges encompass novel efficient PEFT architectures, PEFT for different learning settings, PEFT combined with model compression techniques, and the exploration of PEFT for multimodal LLMs. By presenting this position paper, we aim to stimulate further research and foster discussions surrounding more efficient and accessible PEFT for LLMs.

1 Introduction

Large Language Models (LLMs) have exhibited remarkable capabilities, with popular models such as ChatGPT1 and GPT4 (OpenAI, 2023) showcasing their potentials in a variety of NLP tasks (Zhao et al., 2023; Mohamadi et al., 2023). However, these LLMs often suffer from extensive parameter requirements and associated computational demands, limiting their practicality and scalability for real-world applications. Parameter-Efficient Fine-Tuning (PEFT) addresses the challenges by reducing the number of parameters required for effective fine-tuning without compromising the model performance. Notable PEFT approaches include LoRA (Hu et al., 2022), adapter tuning (Houlsby et al., 2019), prefix-tuning (Li and Liang, 2021), prompt-tuning (Lester et al., 2021), P-tuning (Liu et al., 2022), BitFit (Zaken et al., 2022) and others. Despite the advancement, we observe that existing PEFT approaches for LLMs present several limitations that hinder their effectiveness and practicality. Most PEFT methods are proposed in the BERT era primarily for encoder-based models, which are not tailored specifically for LLMs. Detailed PEFT implementations are mostly agnostic without considering the decoder-only architectures and the algorithmic characteristics of mainstream LLMs, for example, the requirements of Reinforcement Learning from Human Feedback (RLHF)-based fine-tuning (Ouyang et al., 2022). Thus, there is an urgent need for better PEFT that enables more effective learning of LLMs.

In this position paper, we advocate for the development of PEFT techniques specifically tailored for LLMs. We briefly review current states of development in the field. Based on our empirical study, we show that in general LoRA-based approaches are more suitable for LLMs; yet there are no uniform algorithmic designs for all the settings. In addition, we discuss complicated learning strategies that are not supported by current PEFT methods, such as more efficient distributed PEFT, PEFT that support RLHF training for better human alignment, PEFT combines with various model compression techniques (such as distillation and quantization), and PEFT for multi-modal LLMs. We hope that our research can stimulate research for better PEFT techniques, especially for LLMs.

2 Literature Review

A Brief Overview of LLMs. Before the LLM wave, Pre-trained Language Models (PLMs) have gained significant attention due to their abilities to learn contextual representations (Qiu et al., 2020; Min et al., 2021). One prominent example is BERT (Devlin et al., 2019), which leverages the encoder-only architecture and has been adopted in language understanding tasks. Since the launch

https://openai.com/blog/chatgpt

of ChatGPT, a variety of LLMs have been released. Popular open LLMs include LLaMA (Touvron et al., 2023a), LLaMA 2 (Touvron et al., 2023b), OPT (Zhang et al., 2022), OPT-IML (Iyer et al., 2022), GPT-NeoX (Black et al., 2022), BLOOM (Scao et al., 2022), BLOOMZ (Muennighoff et al., 2023), Galactica (Taylor et al., 2022), CPM-2 (Zhang et al., 2021), GLM (Zeng et al., 2023), Pythia (Biderman et al., 2023), and many others, to name a few. For model training, the three stage process of "pre-training, supervised finetuning (SFT) and RLHF" put forward by (Ouyang et al., 2022) is widely accepted by the community. It can be easily seen that training LLMs requires numerous computational resources. Therefore, the huge computational and financial costs naturally call for the development of PEFT for LLMs.

General PEFT Methods. PEFT is a type of finetuning methods that reduce the number of learnable parameters of PLMs (not specifically for LLMs) while preserving good performance, which is also referred to as Delta Tuning (Ding et al., 2023). Bit-Fit (Zaken et al., 2022) is a simple sparse finetuning method where only the bias parameters are tuned. LoRA (Hu et al., 2022) leverages lowrank approximation to the update matrices (i.e., parameters) at each model layer, which can be applied to various PLMs. Following the work of LoRA, AdaLoRA (Zhang et al., 2023a) is proposed to incorporate adaptive budget allocation into the choices of LoRA ranks for different matrices. Dy-LoRA (Valipour et al., 2023) further employs a dynamic search-free technique for rank selection. Adapters (Houlsby et al., 2019) are small neural network modules integrated into original transformer blocks, which are learned to capture new knowledge for downstream tasks. AdaMix (Wang et al., 2022) learns a mixture of multiple adapters for PEFT. Prefix-tuning (Li and Liang, 2021) adds a sequence of prefixes represented as trainable continuous embeddings to each transformer layer that specifically capture the task-specific information. Adaptive Prefix-tuning (Zhang et al., 2023c) extends Prefix-tuning to make the lengths of prefixes more adaptive to tasks. P-tuning v2 (Liu et al., 2022) is a similar approach that shows layerwise prompt vectors are also beneficial for solving language understanding tasks. Prefix Propagation (Li et al., 2023) explores prefix-tuning for longer input sequences. In contrast to continuous vectors, prompt-tuning employs trainable prompt vectors (Lester et al., 2021; Liu et al., 2021; Wang

et al., 2021; Xu et al., 2023b) or discrete textual descriptions (Shin et al., 2020; Gao et al., 2021) at the input layer to model task-level knowledge. We refer reader to the survey (Liu et al., 2023b) for a more detailed review.

PEFT Methods for LLMs. It is worth noting that the above methods are not tailored to LLMs. Thus, we further summarize how these PEFT techniques are applied. To the best of our knowledge, LoRA (Hu et al., 2022) is one of the most widely applied methods due to its simplicity in design and uniformity in application scenarios. Apart from LoRA, LLaMA-Adapter (Zhang et al., 2023b) is proposed to insert adapter networks into LLMs with zero-initialized attention. In the open-source community, PEFT² is also the name of a project that provides the implementations of several PEFT methods on LLMs, serving as a useful tool for further research into the subject. OpenDelta (Hu et al., 2023) focuses on the quick adaptation of LLMs. A few works focus on evaluating PEFT on text generation tasks (Chen et al., 2022; Xu et al., 2022), but are not conducted for LLMs. Another thread of works combine model quantization with PEFT, which maps model parameters from floatingpoint numbers to integers (Gholami et al., 2021). QLoRA (Dettmers et al., 2023) quantizes an LLM to 4-bit, and then leverages a small set of LoRA weights to avoid performance degradation. Alpha Tuning (Kwon et al., 2022) and QA-LoRA (Xu et al., 2023a) are quantization-aware adaptation methods for LLMs. AWO (Lin et al., 2023) significantly reduces the model quantization error by protecting 1% of the salient weights of the LLM.

3 Analysis and Research Directions

We analyze the performance of PEFT on LLMs and suggest several directions for future research.

3.1 Empirical Analysis

Before presenting research directions, we conduct a brief empirical analysis on the effectiveness of PEFT over LLMs. Without loss of generality, we evaluate the performance of a popular LLM, i.e., Llama-2-7b-chat³, over two text generation tasks (E2E (Novikova et al., 2017) and WebNLG (Gardent et al., 2017) and two more challenging task of math problems (GSM8K (Cobbe et al., 2021) and

²https://github.com/huggingface/peft
3https://huggingface.co/meta-llama/
Llama-2-7b-chat-hf

Metric	FT	LoRA	Prompt	Prefix	
Dataset: E2E (Text Generation)					
BLEU-1	0.5460	0.5000	0.4476	0.4552	
BLEU-2	0.3956	0.3486	0.2994	0.3252	
METEOR	0.3448	0.3265	0.2816	0.2952	
ROUGE-L	0.3918	0.3569	0.3153	0.3312	
CIDEr	0.9502	0.7646	0.5163	0.6003	
Dataset: WebNLG (Text Generation)					
BLEU-1	0.3025	0.3217	0.2352	0.2587	
BLEU-2	0.2109	0.2173	0.1943	0.2005	
METEOR	0.2014	0.1992	0.1698	0.1754	
ROUGE-L	0.3045	0.2881	0.2398	0.2465	
CIDEr	0.6207	0.5029	0.3465	0.4186	
Dataset: GSM8K (Math Problem)					
Accuracy	0.2382	0.21542	0.15643	0.17454	
Dataset: CoQA (Question Answering)					
EM	0.6089	0.5976	0.5193	0.5339	
F1	0.7004	0.6958	0.5930	0.6264	

Table 1: Performance of PEFT methods and FT over multiple generation tasks. Note: FT (full fine-tuning), Prompt (prompt-tuning), Prefix (prefix-tuning).

question answering (CoQA (Reddy et al., 2019)). Detailed dataset statistics and experimental settings can be found in the appendix.

In Table 1, we report the testing performance of standard fine-tuning and three popular PEFT methods, namely, prompt-tuning (Lester et al., 2021), prefix-tuning (Li and Liang, 2021) and LoRA (Hu et al., 2022). Results show that LoRA and full finetuning exhibit similar performance in generative tasks, with minimal differences in quality of generated contents. For math problems and QA tasks, they display a slight variance in accuracy, whereas other PEFT methods perform inadequately. In Table 2 we study the effectiveness of LoRA on a larger model scale based on Llama-2 and Vicuna⁴. The results indicate that for simple text generation, there is a minor enhancement as the model size increases from 7B to 13B. However, there is no discernible difference in the generation quality of specific questions after manual checking. Conversely, for more intricate math problems, we observe a significant improvement in accuracy with the increase in model parameters.

We further observe that different LoRA ranks have varying degrees of performance impact. To investigate this further, we conduct tests on the same dataset with different data volumes (randomly sampled 5%, 10%, and the entire dataset) using different LoRA ranks. As shown in Figure 1, for smaller datasets, a lower LoRA rank yields optimal results, and increasing LoRA ranks actually leads

Metric	Llama-2 (7B)	Llama-2 (13B)	Vicuna (7B)	Vicuna (13B)	
Dataset: E2E (Text Generation)					
BLEU-1	0.5000	0.5028	0.5038	0.5066	
BLEU-2	0.3486	0.3522	0.3526	0.3545	
METEOR	0.3265	0.3228	0.3238	0.3247	
ROUGE-L	0.3569	0.3559	0.3558	0.3560	
CIDEr	0.7646	0.7986	0.7783	0.8106	
Dataset: GSM8K (Math Problem)					
Accuracy	0.2382	0.3835	0.1641	0.2273	

Table 2: LoRA performance with different model sizes.

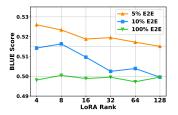


Figure 1: The impact of data volume (5%, 10%, 100%) of the E2E dataset) with different LoRA ranks.

to a decline in performance. Therefore, a lower LoRA rank can achieve satisfactory performance while also saving training resource costs.

3.2 Lessons Learned for Future Research

From the experiments, it is seen that LoRA-style PEFT methods achieve better performance for LLMs. Yet, there is no "free lunch" for all learning settings, particularly for different tasks and data volumes. In addition, the trained LoRA modules with large ranks may still be over-parameterized for some cases. We suggest that for future research, here are some possible directions. i) Task-adaptive LoRA methods can be developed to search more suitable ranks based on task difficulty and data volumes. ii) More compact low-rank structures can be involved to decompose the parameter matrices, which speed up the training process and avoid overfitting simultaneously. iii) Combining LoRA-style approaches with better prompt designs for LLMs may also result in better performance.

3.3 Other Research Directions

Large-scale Training. As observed from the experimental results, the performance of LoRA is highly related to the number of trainable parameters (controlled by the LoRA rank), for LLMs with 100B parameters or more (such as GPT-4 (OpenAI, 2023)), even turning only 1% of the parameters leads to huge computational costs. In addition, the model checkpoints must be partitioned as they do not fit in single GPU. Thus, the parameters LoRA of modules are also distributed according to the model

⁴https://lmsys.org/blog/ 2023-03-30-vicuna/

partition strategies during training. The parameter values should be communicated frequently across GPUs and machines during the training process. To the best of our knowledge, there are no comprehensive studies or publicly available frameworks that address the problems of large-scale, distributed LoRA training for ultra-large models effectively.

In addition, the auto-regressive language model next token prediction objective is not the only learning task during the LLM training process. For better alignment with human values, RLHF (Ouyang et al., 2022) is often leveraged to fine-tune the LLMs based on reinforcement learning coached by a reward model. This process is more computationally expensive due to the involvement of both the supervised fine-tuned and RLHF-based fine-tuned LLM checkpoints, together with a reward model that expresses human preferences. Compared to simple fine-tuning, RLHF requires the computational graphs and weights of these additional models loaded into the GPU memory during training, which significantly lowers the GPU memory space for training the LLM itself. We suggest that further studies on PEFT-style RLHF training is of greater value to save computational resources and benefit the NLP community for deeper research into how to apply RLHF more easily.

PEFT with Model Compression. For application developers, it is more important to deploy LLMs online for real-time inference. Hence, compressing LLMs to smaller sizes is critical, in order to save GPU memory and speedup the inference process. In the literature, several types of approaches have been proposed to compress the models, such as knowledge distillation, model quantization and pruning. Take quantization as example. In QLoRA (Dettmers et al., 2023), the underlying LLM is quantized to 4-bit first and then tuned using LoRA over a small but high-quality dataset. The work (Hsieh et al., 2023) distills LLMs by extracting rationales as additional supervision from larger models for training small models, yet the parameters of small models require to be fully fine-tuned to ensure high performance. LLM-Pruner (Ma et al., 2023) leverages structural pruning for LLMs to selectively removes non-critical structures based on the gradients learned during training. A similar work Wanda (Sun et al., 2023) prunes weights smallest magnitudes multiplied by the corresponding input activations, in order to bring parameter sparsity to large models. We suggest that the research on LLM compression with PEFT is vital

for online deployment and highly insufficient in current states. For example, it is possible to obtain a smaller model by PEFT-applied distillation. This benefits institutions or developers where fully fine-tuning smaller models (with parameters around 7B) is computationally prohibitive.

PEFT for Multi-modal LLMs. LLMs are not only about texts. By feeding the output representations of visual encoders (or encoders for other modalities) into LLM backbones, multi-modal LLMs, including NExt-GPT (Wu et al., 2023), Instruct-BLIP (Dai et al., 2023), mPLUG-Owl (Ye et al., 2023), LLaVa (Liu et al., 2023a), MiniGPT (Chen et al., 2023) and many others, can be trained and deployed to tackle multi-modal tasks by instruction following. In multi-modal LLMs, unifying the representations of different modalities into the same semantic space is crucial for multi-modal understanding and generation. For instance, Instruct-BLIP (Dai et al., 2023) leverages a Q-Former to extract instruction-aware visual features as the input to a frozen LLM. However, without the training of the LLM, it obtains no new knowledge on how to solve the multi-modal tasks. We believe that PEFT can act as the "bridge" to achieve cross-modal communications by slightly tuning existing LLMs that effectively prevents the catastrophic forgetting of uni-modal knowledge.

Other Topics. In addition to the above mentioned topics, there are other topics that are worth exploring. Strategies such as adaptive learning rates and regularization methods specifically designed for PEFT can further faster and stabilize the training process. Apart from RLHF, examining parameter-efficient ways to address ethical considerations, such as knowledge securities, fairness or bias mitigation (Fan et al., 2023), can contribute to the development of more reliable and unbiased LLMs. Due to space limitation, we do not elaborate.

4 Concluding Remarks

In this position paper, we have highlighted the pressing need for better PEFT methods tailored for LLMs, underscoring the importance of addressing the challenges and open issues in PEFT and encompassing the exploration of novel efficient PEFT architectures, PEFT for different learning settings, and PEFT for multi-modal LLMs. By addressing these challenges, we can pave the way for more efficient and accessible PEFT techniques that are more practical for real-world applications.

Limitations

This paper is a position paper and does not present any specific new methodologies or approaches that could be employed to tackle the identified challenges. The limitations and research directions mentioned in the paper are based on the authors' perspectives and may not encompass the entire scope of issues related to PEFT for LLMs.

References

- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.
- Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. Gpt-neox-20b: An open-source autoregressive language model. *CoRR*, abs/2204.06745.
- Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. 2022. Revisiting parameter-efficient tuning: Are we really there yet? In *EMNLP*, pages 2612–2626. Association for Computational Linguistics.
- Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *CoRR*, abs/2310.09478.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.
- Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *CoRR*, abs/2305.06500.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *CoRR*, abs/2305.14314.

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT, pages 4171–4186. Association for Computational Linguistics.
- Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nat. Mac. Intell.*, 5(3):220–235.
- Mingyuan Fan, Cen Chen, Chengyu Wang, and Jun Huang. 2023. On the trustworthiness landscape of state-of-the-art generative models: A comprehensive survey. *CoRR*, abs/2307.16680.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *ACL/INCNLP*, pages 3816–3830. Association for Computational Linguistics.
- Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *ACL*, pages 179–188. Association for Computational Linguistics.
- Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A survey of quantization methods for efficient neural network inference. *CoRR*, abs/2103.13630.
- 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 ICML, volume 97 of Proceedings of Machine Learning Research, pages 2790–2799. PMLR.
- Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *ACL* (*Findings*), pages 8003–8017.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net.
- Shengding Hu, Ning Ding, Weilin Zhao, Xingtai Lv, Zhen Zhang, Zhiyuan Liu, and Maosong Sun. 2023. Opendelta: A plug-and-play library for parameter-efficient adaptation of pre-trained models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2023, Toronto, Canada, July 10-12, 2023*, pages 274–281. Association for Computational Linguistics.
- Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster,

- Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Ves Stoyanov. 2022. OPT-IML: scaling language model instruction meta learning through the lens of generalization. *CoRR*, abs/2212.12017.
- Se Jung Kwon, Jeonghoon Kim, Jeongin Bae, Kang Min Yoo, Jin-Hwa Kim, Baeseong Park, Byeongwook Kim, Jung-Woo Ha, Nako Sung, and Dongsoo Lee. 2022. Alphatuning: Quantization-aware parameter-efficient adaptation of large-scale pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3288–3305. Association for Computational Linguistics.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, pages 3045–3059. Association for Computational Linguistics.
- Jonathan Li, Will Aitken, Rohan Bhambhoria, and Xiaodan Zhu. 2023. Prefix propagation: Parameterefficient tuning for long sequences. In ACL, pages 1408–1419. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL/IJCNLP*, pages 4582–4597. Association for Computational Linguistics.
- Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. 2023. AWQ: activation-aware weight quantization for LLM compression and acceleration. *CoRR*, abs/2306.00978.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. *CoRR*, abs/2304.08485.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023b. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):195:1–195:35.
- 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 *ACL*, pages 61–68. Association for Computational Linguistics.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *CoRR*, abs/2103.10385.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*. OpenReview.net.
- Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *CoRR*, abs/2305.11627.

- Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey. *CoRR*, abs/2111.01243.
- Salman Mohamadi, Ghulam Mujtaba, Ngan Le, Gianfranco Doretto, and Donald A. Adjeroh. 2023. Chatgpt in the age of generative AI and large language models: A concise survey. *CoRR*, abs/2307.04251.
- Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *ACL*, pages 15991–16111. Association for Computational Linguistics.
- Jekaterina Novikova, Ondrej Dusek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *SIGdial*, pages 201–206. Association for Computational Linguistics.
- OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.
- Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *CoRR*, abs/2003.08271.
- Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266.
- Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien,

- David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.
- Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *EMNLP*, pages 4222–4235. Association for Computational Linguistics.
- Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2023. A simple and effective pruning approach for large language models. *CoRR*, abs/2306.11695.
- Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *CoRR*, abs/2211.09085.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. CoRR, abs/2307.09288.
- Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2023. Dylora: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *EACL*, pages 3266–3279. Association for Computational Linguistics.
- Chengyu Wang, Jianing Wang, Minghui Qiu, Jun Huang, and Ming Gao. 2021. Transprompt: Towards an automatic transferable prompting framework for few-shot text classification. In *EMNLP*, pages 2792–2802. Association for Computational Linguistics.

- Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022. Adamix: Mixture-of-adaptations for parameter-efficient model tuning. In *EMNLP*, pages 5744–5760. Association for Computational Linguistics.
- Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. 2023. Next-gpt: Any-to-any multimodal LLM. *CoRR*, abs/2309.05519.
- Peng Xu, Mostofa Patwary, Shrimai Prabhumoye, Virginia Adams, Ryan Prenger, Wei Ping, Nayeon Lee, Mohammad Shoeybi, and Bryan Catanzaro. 2022.
 Evaluating parameter efficient learning for generation. In *EMNLP*, pages 4824–4833. Association for Computational Linguistics.
- Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. 2023a. Qa-lora: Quantization-aware low-rank adaptation of large language models. *CoRR*, abs/2309.14717.
- Ziyun Xu, Chengyu Wang, Minghui Qiu, Fuli Luo, Runxin Xu, Songfang Huang, and Jun Huang. 2023b. Making pre-trained language models end-to-end few-shot learners with contrastive prompt tuning. In *WSDM*, pages 438–446. ACM.
- Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023. mplug-owl: Modularization empowers large language models with multimodality. *CoRR*, abs/2304.14178.
- Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *ACL*, pages 1–9. Association for Computational Linguistics.
- Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: an open bilingual pre-trained model. In *ICLR*. OpenReview.net.
- Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023a. Adaptive budget allocation for parameter-efficient fine-tuning. In *ICLR*. OpenReview.net.
- Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. 2023b. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *CoRR*, abs/2303.16199.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher

Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Zhengyan Zhang, Yuxian Gu, Xu Han, Shengqi Chen, Chaojun Xiao, Zhenbo Sun, Yuan Yao, Fanchao Qi, Jian Guan, Pei Ke, Yanzheng Cai, Guoyang Zeng, Zhixing Tan, Zhiyuan Liu, Minlie Huang, Wentao Han, Yang Liu, Xiaoyan Zhu, and Maosong Sun. 2021. CPM-2: large-scale cost-effective pre-trained language models. *AI Open*, 2:216–224.

Zhenru Zhang, Chuanqi Tan, Haiyang Xu, Chengyu Wang, Jun Huang, and Songfang Huang. 2023c. Towards adaptive prefix tuning for parameter-efficient language model fine-tuning. In *ACL*, pages 1239–1248. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

A Datasets and Experimental Settings

Datasets. We evaluate the results on two standard neural generation datasets for the table-to-text task: E2E (Novikova et al., 2017), WebNLG (Gardent et al., 2017), one math problem reasoning dataset: GSM8K (Cobbe et al., 2021) and one QA dataset CoQA (Reddy et al., 2019).

Specifically, the E2E dataset contains approximately 50K examples featuring 8 distinct fields. It includes multiple test references for each source table and has an average output length of 22.9. We employ the official evaluation script, which provides metrics such as BLEU, METEOR, ROUGE-L and CIDEr for assessment. The WebNLG dataset consists of 22K examples where the input x consists of sequences of (subject, property, object) triples. The average output length is 22.5. The training and validation splits encompass input descriptions of entities from 9 distinct DBpedia categories, such as Monuments. The test split is divided into two sections: the first half contains categories observed in the training data, while the second half includes 5 unseen categories for extrapolation evaluation. For evaluation, we also utilize the official evaluation script. GSM8K presents a challenging arithmetic reasoning task that language models frequently find difficult to tackle. CoQA is

Dataset	Epoch	Sequence Length
E2E	10	256
WebNLG	10	256
GSM8K	10	512
CoQA	5	2048

Table 3: Hyper-parameter settings for individual datasets.

	Value
Learning Rate	3e-6
AdamW (β_1, β_2)	(0.9, 0.98)
Dropout	0.1
Weight Decay	0.01
Batch Size	48

Table 4: Hyper-parameter settings for all datasets.

a challenging task to measure the model abilities to understand a text passage and answer a series of related questions.

Experimental Settings. The experiments are conducted on a Linux server with two NVIDIA A100-80G GPUs. We choose Llama-2-7b-chat as the default LLM. In addition, Llama-2-13b-chat, together with the 7B and 13B versions of the Vicuna models, is leveraged for study.

Hyper-parameter Settings. At training time, we use AdamW (Loshchilov and Hutter, 2019) as the optimizer, and set its hyper-parameter $(\beta 1, \beta 2)$ to (0.9, 0.98). The hyper-parameters we tune include the number of epochs, the batch size, the learning rate, and the sequence length. Hyper-parameter details are in shown in Table 3 and Table 4.