GPT vs RETRO: Exploring the Intersection of Retrieval and Parameter-Efficient Fine-Tuning

Aleksander Ficek*, **Jiaqi Zeng***, **Oleksii Kuchaiev**
NVIDIA
{aficek,jiaqiz,okuchaiev}@nvidia.com

Abstract

Parameter-Efficient Fine-Tuning (PEFT) and Retrieval-Augmented Generation (RAG) have become popular methods for adapting large language models while minimizing compute requirements. In this paper, we apply PEFT methods (P-tuning, Adapters, and LoRA) to a modified Retrieval-Enhanced Transformer (RETRO) and a baseline GPT model across several sizes, ranging from 823 million to 48 billion parameters. We show that RETRO models outperform GPT models in zero-shot settings due to their unique pre-training process but GPT models have higher performance potential with PEFT. Additionally, our study indicates that 8B parameter models strike an optimal balance between cost and performance and P-tuning lags behind other PEFT techniques. We further provide a comparative analysis between applying PEFT to an Instruction-tuned RETRO model and base RETRO model. This work presents the first comprehensive comparison of various PEFT methods integrated with RAG, applied to both GPT and RETRO models, highlighting their relative performance.

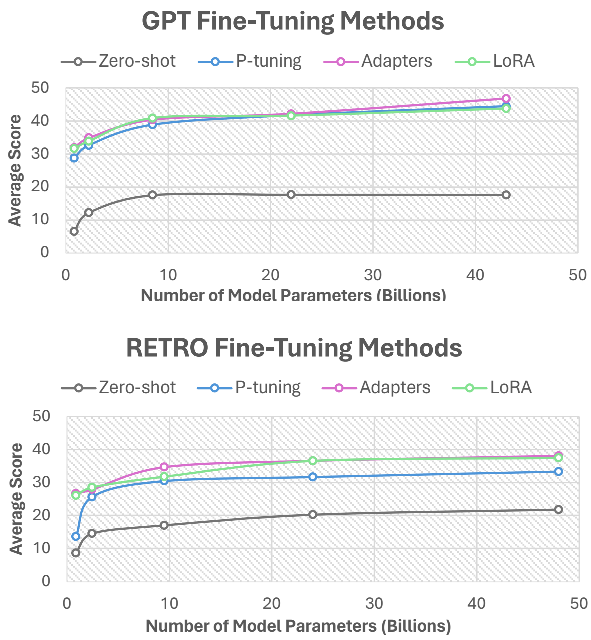

Figure 1: Average GPT vs RETRO scores of six datasets across model sizes of 823M to 48B parameters.

1 Introduction

Pre-trained large language models have made a demonstrable impact across applications in academia and industry. Many use cases, however, require LLMs adapted to specific tasks and unique information but lack the resources for extensive retraining. To address this, Parameter-Efficient Fine-Tuning (PEFT) (Han et al., 2024) and Retrieval-Augmented Generation (RAG) (Gao et al., 2023) have become popular methods due to their effectiveness and efficiency, inspiring new lines of research.

PEFT has been proven to be a comparable substitute to Supervised Fine-Tuning (SFT) by achieving competitive performance at a fraction of the number of updated parameters (Han et al., 2024). In this paper we select P-tuning (Liu et al., 2023), Adapter modules (Houlsby et al., 2019) and Low-Rank Adaptation (LoRA) (Hu et al., 2021) as representative PEFT methods. P-tuning involves training continuous prompt embeddings to guide output for specific tasks without modifying base model parameters. Adapters operate by training fully connected layers inserted throughout the base model while keeping the remaining parameters frozen. LoRA further decomposes the inserted layers into low-rank matrices, enhancing efficiency.

Retrieval-augmented generation (RAG) improves model quality by incorporating external knowledge through mechanisms like BM-25 or TF-IDF (Robertson et al., 2009), online web search (Page et al., 1999), or trained dense retriever models (Karpukhin et al., 2020). Any LLM can be transformed into a retrieval-augmented model by concatenating retrieved sources with the input query,

*Equal contribution.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 19425–19432
November 12-16, 2024

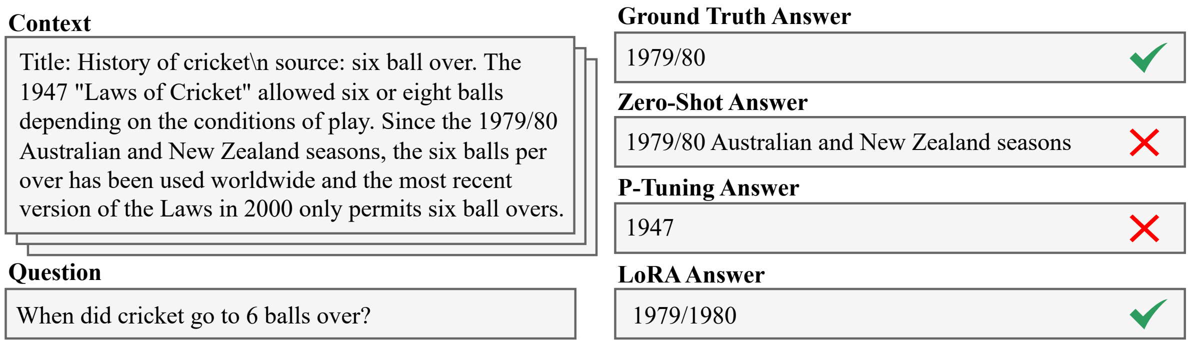

Figure 2: Sample entry inputs and outputs from NQ dataset

provided it fits within the model's context window. Xu et al. (2023) found that retrieval significantly improves GPT model quality on long context tasks, reducing the "lost in the middle" effect (Liu et al., 2024) and offering inherent efficiency benefits.

Alternatively, there exist multiple works (Borgeaud et al., 2022; Guu et al., 2020; Izacard et al., 2022; Nakano et al., 2021) that have integrated retrieval as part of model pretraining or finetuning to notable success when compared to typical GPT models despite being a much lesser explored domain. RETRO (Borgeaud et al., 2022) is of particular interest due to its unique approach of incorporating a retrieval module directly into the transformer architecture via a chunked-cross attention mechanism and ability to scale to trillions of tokens resulting in reduced perplexity. Subsequently, Wang et al. (2023b) showed that RETRO at sizes up to 9.5 billion parameters largely outperforms GPT on specific knowledge-intensive tasks. Furthermore, Wang et al. (2023a) illustrated that when scaled up to 48 billion parameters and instruction-tuned, RETRO performed better than equivalent GPT models on several question answering, reading comprehension and summarization tasks.

In this paper we continue the exploration of RETRO versus GPT through the lens of parameter efficient finetuning. We apply P-tuning, Adapter modules and LoRA to multiple tasks with retrieval for both RETRO and GPT models. To our knowledge, this paper provides the first in-depth comparison of various Parameter Efficient Fine-Tuning integrated with Retrieval-Augmented Generation, uniquely applied to both GPT and RETRO models.

2 Related Work

Previous works like (Chen et al., 2022) have compared multiple PEFT methods but lacked comparison for retrieval-based tasks and retrieval augmented language models. In this section we focus on recent work that combine finetuning with retrieval. A comprehensive survey (Gao et al., 2023) synthetized multiple comparative studies on PEFT and RAG, underscoring the potential benefits of combining these approaches as a promising direction for future investigation. There are multiple works that provide methods to combine RAG with fine-tuning to improve accuracy (Zhang et al., 2024a,b; Rangan and Yin, 2024). Multiple studies have explored the comparison between fine-tuning and retrieval. Lakatos et al. (2024) and Ovadia et al. (2023) reported improved accuracy using RAG over fine-tuning GPT models, while also noting suboptimal results when combining the two methods. Gupta et al. (2024) demonstrated improved outcomes by integrating both approaches for specific agriculture and geography tasks. Additionally, Soudani et al. (2024) compared the efficacy of these methods, including full and QLoRA finetuning (Dettmers et al., 2024), in low-frequency entity question-answering tasks. These studies collectively suggest the need for comprehensive investigation into multiple PEFT techniques combined with RAG and maintain retrieval pretrained LLMs with PEFT to be unexplored, thereby motivating our research.

3 Experimental Setup

3.1 Datasets

To cover several task categories, we use six datasets suited to benefit from retrieval and finetuning. We select **Natural Questions** (NQ) (Kwiatkowski et al., 2019), **TriviaQA** (TQA) (Joshi et al., 2017), **NarrativeQA** (NQA) (Kočiský et al., 2018) and **Qasper** (Dasigi et al., 2021) for document question answering, **QuALITY** (Pang et al., 2021) for multiple-choice question answering, and **QMSum**

(Zhong et al., 2021) for query-based summarization. Table 1 details the sizes of dataset training, validation and test partitions. Each of these datasets contain necessary external knowledge that must be filtered via retrieval and response behaviour that encourages finetuning. Following the official metrics, we use F1 score for evaluating document QA, exact match for mutliple-choice QA and the geometric mean of ROUGE-1/2/L (Lin, 2004) for summarization.

	NQ	TQA	NQA	QASPER	QUALITY	QMSUM
Train	79168	78785	44002	2053	2018	1005
Valid	8757	8837	11001	514	505	252
Test	3610	11313	5859	1726	2086	272

Table 1: Number of samples in train/validation/test split for each dataset.

3.2 Models

In order to understand the effect of model scaling, we use base GPT models of sizes 823M (Extra Small), 2.25B (Small), 8.5B (Medium), 22B (Large), and 43B (Extra Large), as introduced in Wang et al. (2023a), which were pretrained on a massive dataset of 1.2 trillion tokens. We employ the corresponding RETRO models from the same work as the foundation for our retrieval pretrained LLM experiments. Notably, the RETRO architecture features an encoder that extracts neighbors from an external database, which increases the total model size to 877M, 2.47B, 9.5B, 24B, and 48B, respectively. Wang et al. (2023a) found ablating the encoder after pretraining led to comparable results. In our paper we include it so that adapter modules and LoRA layers are added throughout decoder and encoder components. We choose the GPT and RETRO model types for our experiments because they are representative architectures of the general and retrieval LLM landscape while allowing us to leverage the large pretrained models introduced in Wang et al. (2023a). For more information on the base models we refer readers to the original work.

3.3 Retrieval

We follow Wang et al. (2023a); Xu et al. (2023) to use Dragon+ (Lin et al., 2023) as a retriever. Dragon+ is a dual encoder model that consists of a query encoder and a context encoder. We first chunk each context document with 100 words, and then encode both the questions and all chunks independently with corresponding encoders. The most relevant 5 chunks, ranked by the dot product of the question embedding and chunk embedding, are retrieved as neighbors. For GPT models, they are concatenated together (following the left to right order from the most relevant to least relevant) as the context of the prompt for generation. For RETRO models, they interact with the question during generation through chunked cross-attention. We choose Dragon+ as the retriever because it was employed in the original RETRO paper (Borgeaud et al., 2022) and has achieved decent performance in other works (Wang et al., 2023a). Here we are interested in relative performance between GPT and RETRO models, enabling comparison against the architectures instead of comparing multiple retrievers which we leave for future work.

3.4 Parameter Efficient Fine-Tuning

We implement P-tuning in RETRO akin to GPT. Virtual tokens are added to the beginning of the decoder. Based on the design of chunked cross-attention, left padding is added to ensure the length of input (virtual tokens + context + question) is a multiple of chunk size. Adapter and LoRA layers are in all attention layers in both transformer architectures. This means that for RETRO they are also inserted in the retrieval encoder which receives retrieved neighbors. We provide additional hyperparameter tuning, resource utilization and prompt template details in Appendix A. We also include Table 2 for a full list of base and PEFT model parameter counts.

Type	Size	Base Model	P-Tuning	Adapters	LoRA
GPT	Extra Small	823M	2.2M	3.2M	3.1M
	Small	2.25B	3.3M	6.5M	6.3M
	Medium	8.5B	5.6M	18.8M	16.8M
	Large	22B	8.0M	35.2M	31.2M
	Extra Large	43B	10.4M	63.2M	50.4M
RETRO	Extra Small	877M	2.2M	3.6M	4.3M
	Small	2.47B	3.3M	7.3M	8.7M
	Medium	9.5B	5.6M	20.8M	22.4M
	Large	24B	8.0M	43.5M	42.4M
	Extra Large	48B	10.4M	70.6M	68.0M

Table 2: Base and PEFT model number of parameters

4 Results

4.1 Main Results

Table 3 shows the comprehensive comparison between GPT and RETRO models across five model sizes and six datasets. We perform zero-shot and PEFT on all cases and fine-tuning on small and medium model sizes. From this table we observe:

		NQ		TQA		NQA		QASPER		QUALITY		QMSUM		AVERAGE	
		GPT	RETRO	GPT	RETRO	GPT	RETRO	GPT	RETRO	GPT	RETRO	GPT	RETRO	GPT	RETRO
Extra Small	Zero-shot	2.95	**8.28**	9.99	**19.26**	**7.07**	4.87	9.00	**10.79**	0.38	0.48	**9.83**	7.78	6.54	**8.58**
	P-tuning	**24.74**	7.60	**63.63**	24.61	**16.74**	6.69	**24.54**	11.94	**24.59**	17.45	**18.25**	13.63	**28.75**	13.65
	Adapter	**38.48**	23.69	**67.99**	59.60	**17.76**	15.42	**23.52**	20.96	24.26	**25.93**	**19.74**	14.42	**31.96**	26.67
	LoRA	**37.09**	22.13	**67.31**	59.02	**18.08**	15.81	**23.54**	19.85	24.93	**25.65**	**19.27**	13.79	**31.70**	26.04
Small	Zero-shot	11.65	**18.77**	29.88	**38.42**	7.07	**7.12**	12.31	**12.42**	0.00	**1.01**	**12.35**	9.25	12.21	**14.50**
	P-tuning	**39.27**	18.58	**70.31**	61.13	**19.98**	15.13	**24.75**	20.34	22.77	**24.11**	**18.76**	14.61	**32.64**	25.65
	Adapter	**42.29**	23.68	**73.21**	64.91	**21.40**	18.10	**27.29**	20.55	24.93	**25.07**	20.17	15.03	**34.88**	27.89
	LoRA	**39.27**	28.06	**72.34**	64.59	**20.98**	17.90	**24.83**	21.28	**25.79**	24.69	**20.31**	14.46	**33.92**	28.50
	Fine-tuning	**36.27**	21.87	**73.83**	63.05	**17.80**	13.11	**30.84**	21.26	**26.08**	25.79	**20.79**	14.79	**34.27**	26.65
Medium	Zero-shot	23.67	**24.11**	51.00	**52.17**	**8.90**	6.39	9.01	**10.04**	**1.44**	0.14	**11.28**	9.15	**17.55**	17.00
	P-tuning	**45.52**	24.18	**77.00**	67.94	**24.50**	19.02	**33.31**	24.20	**32.74**	31.93	**20.37**	15.40	**38.91**	30.44
	Adapter	**46.71**	43.01	**78.05**	71.35	**24.30**	20.51	**32.53**	25.90	**40.84**	31.98	**20.03**	15.61	**40.41**	34.65
	LoRA	**46.81**	42.11	**78.26**	70.75	**25.17**	20.42	**31.84**	24.48	**41.56**	32.41	**21.47**	15.30	**40.85**	34.24
	Fine-tuning	**41.34**	29.79	**79.82**	68.84	**22.33**	19.37	**49.67**	23.53	**37.01**	33.56	**21.95**	15.29	**42.02**	31.73
Large	Zero-shot	25.37	**31.43**	48.68	**60.30**	**13.92**	7.98	8.73	**10.52**	**2.97**	1.87	6.30	**9.33**	17.66	**20.24**
	P-tuning	**45.20**	15.78	**78.33**	73.22	**25.21**	21.58	**34.24**	24.50	**47.65**	39.93	**20.07**	15.00	**41.78**	31.67
	Adapter	**47.48**	44.43	**79.68**	73.57	**26.37**	22.03	**32.12**	26.09	**46.74**	38.06	**20.81**	15.22	**42.20**	36.57
	LoRA	**47.33**	44.48	**79.79**	73.63	**25.85**	21.49	**32.25**	25.21	**42.62**	39.31	**21.67**	15.02	**41.58**	36.53
Extra Large	Zero-shot	26.97	**33.49**	44.71	**62.87**	**11.89**	10.07	11.58	**13.38**	**3.07**	0.96	7.65	**9.99**	17.65	**21.79**
	P-tuning	**47.27**	24.53	**80.27**	74.38	**27.09**	22.48	**34.08**	24.93	**57.19**	38.06	**21.17**	15.53	**44.51**	33.32
	Adapter	**49.68**	46.41	**81.64**	75.10	**26.94**	22.24	**33.94**	26.38	**54.65**	42.62	**21.19**	15.71	**46.82**	38.08
	LoRA	**49.21**	44.53	**81.87**	74.92	**27.31**	22.16	**31.98**	27.49	**49.19**	39.65	**22.77**	15.73	**43.72**	37.41

Table 3: A comprehensive comparison between GPT vs RETRO on six datasets. **Bold** indicates the better result in each head-to-head comparison.

1) RETRO is better than GPT at zero-shot retrieval tasks. This superiority stems from its unique pre-training approach and focus on retrieval tasks. By learning to extract salient information from retrieved text and integrate it into its generation process, RETRO develops the capability to harness relevant contextual knowledge, ultimately leading to its strong zero-shot performance. In contrast, GPT relies on an auto-regressive loss during pre-training, focusing on accurately predicting next tokens without the benefit of external retrievals. As a result, GPT's ability to learn context-aware question-answering is limited to the presence of relevant data within the pre-training corpus, resulting in less targeted training compared to RETRO.

2) Both RETRO and GPT models exhibit saturation points around 8B parameters. Additionally, a similar pattern emerges between the two models as they are scaled, albeit with RETRO performing less well. This can be seen in Figure 1 and suggests that, for a specific task, a medium-sized PEFT model strikes the optimal balance between cost and performance, making it a sweet spot for many applications.

3) P-tuning underperforms LoRA and Adapters in smaller GPT models but bests them in larger sizes. This difference is visualized in Figure 3 and Figure 3 (Appendix B). However, for RETRO models, P-tuning generally under performs the other PEFT methods across all model sizes. We believe that P-Tuning's lower parameter count contributes to its lower performance especially when paired with smaller base model sizes. For RETRO specifically P-Tuning We hypothesis that P-Tuning's weaker ability in all RETRO model sizes could lie in architecture differences. In P-tuning, virtual tokens are intentionally prepended to the decoder's input, but they are not included in the retrieval encoder. Although they can influence the encoder through cross-attention, the impact might not be as direct or substantial as required. Alternatively, LoRA and Adapters are added to both encoder and decoder which explains their improved capabilities.

4) The performance ceiling for PEFT-tuned models is notably higher for GPT than RETRO. This is demonstrated in Figure 4 (Appendix B) where example, using medium-sized models, the average score of LoRA with GPT is 40.85, while with RETRO it is 34.24. This disparity suggests that GPT has more room for improvement with PEFT tuning. This phenomenon can also be possibly explained by the two different pre-training strategies. Since GPT pre-training is not focused on retrieval-augmented generation, it opens larger room for improvement during fine-tuning.

5) Full fine-tuning marginally outperforms PEFT in GPT models and underperforms in RETRO models. We find that full fine-tuning in GPT models achieves slightly better performance than PEFT on 4 out of 6 tasks while RETRO slightly underperforms on 5 out of 6 tasks. Interestingly, NQ and NQA underperforms against PEFT in both GPT and RETRO 2B and 8B model sizes while both model sizes see notable improvements in fine-tuning GPT on the QASPER dataset. This aligns with previous findings (Hu et al., 2021), potentially because PEFT serves as a regularization, forcing models to learn better.

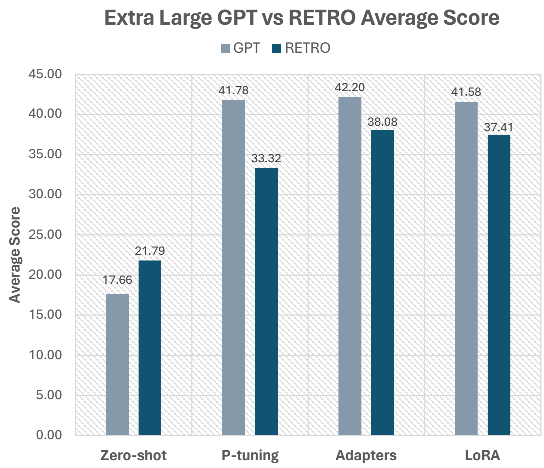

Figure 3: Comparison of Extra Large GPT and RETRO results averaged across 6 datasets.

4.2 Failure Case Analysis

To better frame and qualitatively understand our results we study on an entry from the NQ test set evaluated with Extra-Small RETRO model. Figure 2 demonstrates how zero-shot RETRO is capable of achieving the correct answer but incorrectly formatting the output. Contrarily, P-Tuning incorrectly hallucinates an answer of "1947", the first date seen in the context. LoRA achieves the desired answer by correctly parsing the context and formatting with the desired brevity.

4.3 Comparing to Instruction-tuned RETRO

Instruction tuning post retrieval-augmented pre-training (Wang et al., 2023a) has been demonstrated to improve zero-shot performance on RETRO models. A natural thought is that whether Instruction-tuned RETRO (I-RETRO) serve as a better foundation for applying PEFT compared to the base RETRO. To investigate this, we additionally apply PEFT to a medium-sized I-RETRO model and show overall results in Table 4 and more granular results in Table 5 (Appendix B). Our findings reveal that while I-RETRO exhibits improved performance in the zero-shot setting, it has limited scope for further improvement using PEFT. Even with substantial hyperparameter tuning, the average scores across six datasets, using each of the three PEFT methods, demonstrate an approximately 10% gap between I-RETRO and base RETRO. We hypothesize that conceptually both models should be tunable to similar performance but will leave that exploration to future work.

	Average QA		QUALITY		QMSUM		Average	
	I-RETRO	RETRO	I-RETRO	RETRO	I-RETRO	RETRO	I-RETRO	RETRO
Zero-shot	**27.65**	23.79	**3.35**	0.14	**11.04**	9.15	**20.83**	17.00
P-tuning	23.25	**47.18**	16.68	**31.93**	**15.88**	15.40	20.75	**30.44**
Adapter	22.64	**52.75**	29.87	**31.98**	15.06	**15.16**	22.58	**34.65**
LoRA	26.53	**52.80**	24.21	**32.41**	**15.40**	15.30	24.29	**34.24**

Table 4: Instruction-tuned RETRO evaluation results.

5 Conclusion

This study explores Parameter-Efficient Fine-Tuning (PEFT) methods applied to Retrieval-Augmented Generation (RAG) models, comparing GPT and RETRO architectures. RETRO generally outperforms GPT in zero-shot settings due to their pre-training process that integrates external retrieval, enhancing contextual understanding. However, GPT models show a higher performance potential with PEFT, indicating more room for improvement during fine-tuning. Both RETRO and GPT models perform optimally around the 8B parameter mark, balancing cost and performance. While P-tuning is effective in larger models, it lags behind other methods in smaller models, particularly for RETRO. Applying PEFT to Instruction-tuned RETRO yields limited improvement compared to base RETRO, suggesting a saturation point in leveraging pre-training and fine-tuning benefits. Our comprehensive analysis offers valuable insights for optimizing large language models with PEFT and RAG to the community.

Limitations

Due to the breadth of experiments covered in this work we had to prioritize certain experiments over others. This resulted in us using only the small and medium sized GPT and RETRO models for additional finetuning and Instruction tuning experiments. We believe these results generalize to the other model sizes but leave that to be validated in future work.

Potential Risks

The environmental impact associated with training and fine-tuning large models is not negligible as it involves substantial computational resources and energy consumption. While PEFT aims to alleviate this by reducing the number of tunable parameters, works like ours still require significant compute to distinguish which methods are more promising.

References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. 2022. Revisiting parameter-efficient tuning: Are we really there yet? *Preprint*, arXiv:2202.07962.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. *arXiv preprint arXiv:2105.03011*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Aman Gupta, Anup Shirgaonkar, Angels de Luis Balaguer, Bruno Silva, Daniel Holstein, Dawei Li, Jennifer Marsman, Leonardo O Nunes, Mahsa Rouzbahman, Morris Sharp, et al. 2024. Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture. *arXiv preprint arXiv:2401.08406*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

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 *International conference on machine learning*, 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. *arXiv preprint arXiv:2106.09685*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Atlas: Few-shot learning with retrieval augmented language models. *Preprint*, arXiv:2208.03299.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Robert Lakatos, Peter Pollner, Andras Hajdu, and Tamas Joo. 2024. Investigating the performance of retrieval-augmented generation and fine-tuning for the development of ai-driven knowledge-based systems. *arXiv preprint arXiv:2403.09727*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *AI Open*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback, 2021. *URL https://arxiv. org/abs/2112.09332*.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2023. Fine-tuning or retrieval? comparing knowledge injection in llms. *arXiv preprint arXiv:2312.05934*.

Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd, et al. 1999. The pagerank citation ranking: Bringing order to the web.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, et al. 2021. Qual-ity: Question answering with long input texts, yes. *arXiv preprint arXiv:2112.08608*.

Keshav Rangan and Yiqiao Yin. 2024. A fine-tuning enhanced rag system with quantized influence measure as ai judge. *arXiv preprint arXiv:2402.17081*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. 2024. Fine tuning vs. retrieval augmented generation for less popular knowledge. *arXiv preprint arXiv:2403.01432*.

Boxin Wang, Wei Ping, Lawrence McAfee, Peng Xu, Bo Li, Mohammad Shoeybi, and Bryan Catanzaro. 2023a. Instructretro: Instruction tuning post retrieval-augmented pretraining. *arXiv preprint arXiv:2310.07713*.

Boxin Wang, Wei Ping, Peng Xu, Lawrence McAfee, Zihan Liu, Mohammad Shoeybi, Yi Dong, Oleksii Kuchaiev, Bo Li, Chaowei Xiao, et al. 2023b. Shall we pretrain autoregressive language models with retrieval? a comprehensive study. *arXiv preprint arXiv:2304.06762*.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*.

Liang Zhang, Katherine Jijo, Spurthi Setty, Eden Chung, Fatima Javid, Natan Vidra, and Tommy Clifford. 2024a. Enhancing large language model performance to answer questions and extract information more accurately. *arXiv preprint arXiv:2402.01722*.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024b. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. 2021. Qmsum: A new benchmark for query-based multi-domain meeting summarization. *arXiv preprint arXiv:2104.05938*.

A Details on Experimental Setup

A.1 Hyperparameter Tuning

Given the massive number of experiments required for this work, we used an initial search of learning rates 1e-4 and 1e-5 followed by selectively modifying certain hyperparameters if a model, method and dataset combination did not converge. For all experiments we used a micro batch size of 1 and global batch size of 32 or 128 using tensor parallelism combined with a max sequence length of 1024 and 5 retrieved neighbors. For P-Tuning we selected 100 virtual tokens, kept dropout at 0.0 and used 2 multilayer perceptron layers with hidden sizes of 2048 as the prompt encoder. For Adapters/LoRA we used 32 and 64 dimensions with parallel type adapters and kept dropout at 0.0. In certain runs on NQ and TQA datasets we noticed the models did not converge. To address this, we conducted additional hyperparameter search by varying the learning rates between 1e-4 and 1e-6, testing P-Tuning with 40, 50, and 90 virtual tokens, and selecting Adapters/LoRA with a dimension of 16.

A.2 Resource Utilization

In our experiments, we used up to 16 compute nodes, each with 8 A100-80GB SXM GPUs. When model is smaller, we increased the data parallelism size, using tools in NeMo framework.

A.3 Prompt Template

The template we used to present context to GPT models is as follows.

```
title: {title}
source: {source}
title: {title}
source: {source}
title: {title}
source: {source}
title: {title}
source: {source}
title: {title}
source: {source}

Question: {question} Answer: The answer is
```

B Supplementary Figures and Tables

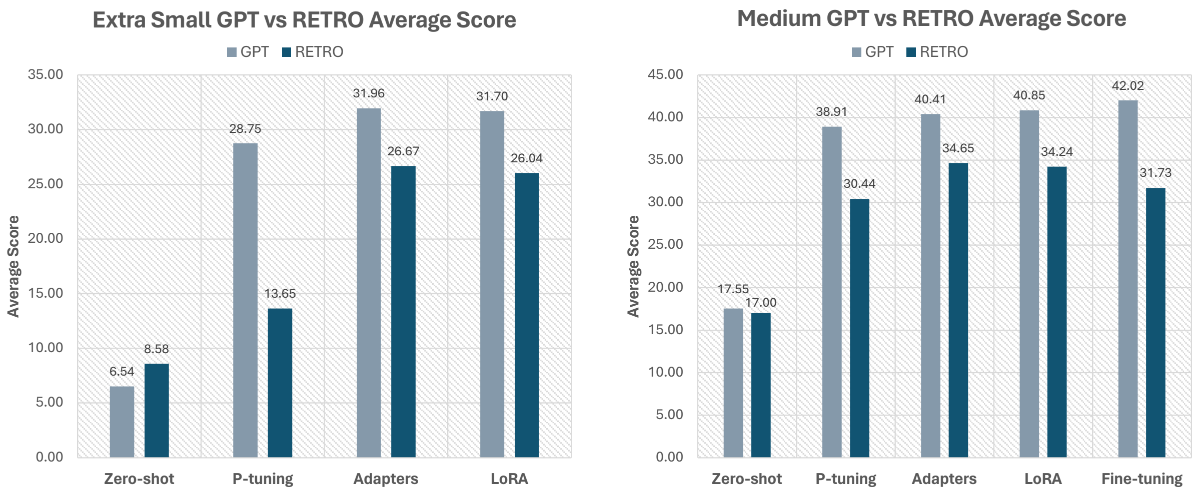

Figure 3: GPT vs RETRO comparisons on Extra Small and Medium sized models.

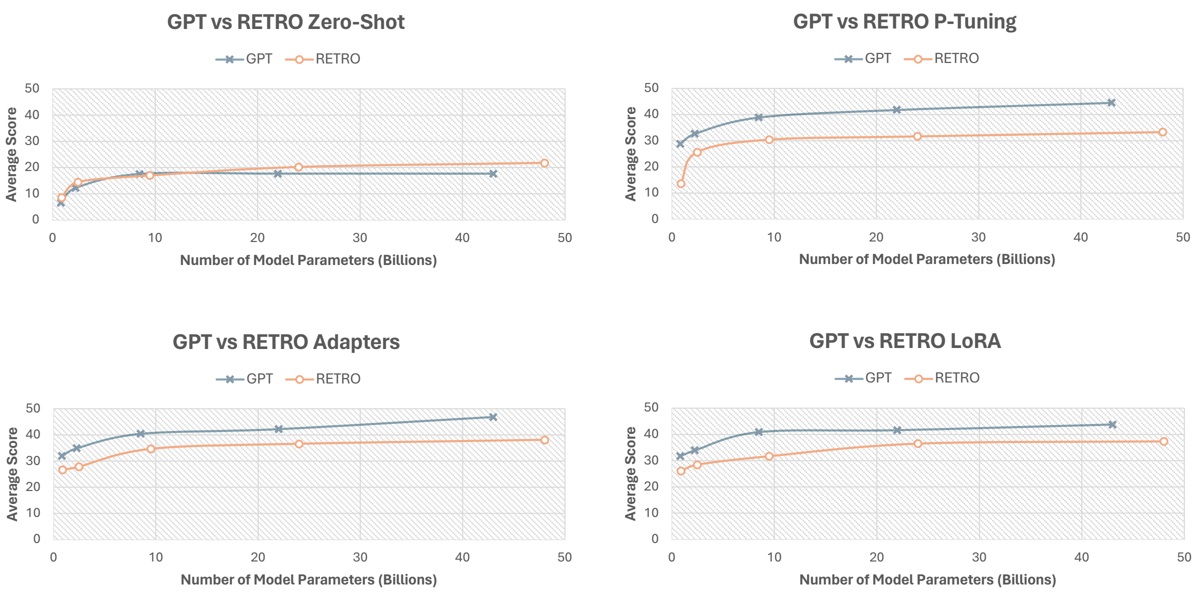

Figure 4: GPT vs RETRO seperate method comparisons.

	NQ		TQA		NQA		QASPER		QUALITY		QMSUM		Average	
	I-RETRO	RETRO	I-RETRO	RETRO	I-RETRO	RETRO	I-RETRO	RETRO	I-RETRO	RETRO	I-RETRO	RETRO	I-RETRO	RETRO
Zero-shot	**30.39**	24.11	**53.25**	52.17	**12.23**	6.39	**14.72**	10.04	**3.35**	0.14	**11.04**	9.15	**20.83**	17.00
P-tuning	19.55	**24.18**	41.95	**67.94**	**20.17**	19.02	11.34	**24.20**	16.68	**31.93**	**15.88**	15.40	20.75	**30.44**
Adapter	18.81	**43.01**	38.83	**71.35**	20.30	**20.51**	12.64	**25.90**	29.87	**31.98**	15.06	**15.16**	22.58	**34.65**
LoRA	21.56	**42.11**	47.89	**70.75**	19.23	**20.42**	17.45	**24.48**	24.21	**32.41**	**15.40**	15.30	24.29	**34.24**

Table 5: Full results with Instruction-tuned RETRO. **Bold** indicates the better result in each head-to-head comparison.