MPLoRA: Orthogonal Multi-Path Low-Rank Adaptation for Parameter Efficient Fine-Tuning

Junhan Shi*
Tsinghua Shenzhen International Graduate School
Shenzhen, China
shijh23@mails.tsinghua.edu.cn

Fulin Wang*
Tsinghua Shenzhen International Graduate School
Shenzhen, China
wfl22@mails.tsinghua.edu.cn

Qing Li
Peng Cheng Laboratory
Shenzhen, China
liq@pcl.ac.cn

Yong Jiang†
Tsinghua Shenzhen International Graduate School
Peng Cheng Laboratory
Shenzhen, China
jiangy@sz.tsinghua.edu.cn

Abstract

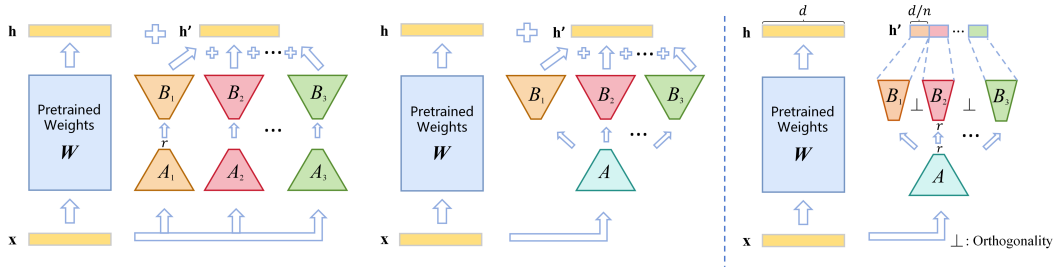
Parameter-efficient fine-tuning (PEFT) has become crucial for adapting large language models to specific tasks, with Low-Rank Adaptation (LoRA) emerging as a prominent method. However, capturing diverse representations within LoRA's limited parameter space remains challenging. We propose Multi-Path LoRA (MPLoRA), a novel approach that decomposes the adaptation matrix into multiple smaller matrices with orthogonal constraints. MPLoRA encourages diverse representations and improves adaptation capability without increasing parameter count. Experiments on various tasks demonstrate that MPLoRA outperforms LoRA and other baselines, with notable improvements on datasets with limited samples. Our analysis reveals that both the multi-path structure and orthogonal constraints contribute significantly to MPLoRA's effectiveness. These findings highlight MPLoRA's potential for enhancing LLM performance and generalization, especially in resource-constrained scenarios, offering new insights into parameter-efficient fine-tuning.

1 Introduction

Recent advancements in machine learning, particularly the development of large language models (LLMs), have led to remarkable performance across diverse domains [Brown et al., 2020]. Nevertheless, their immense size presents significant challenges for task-specific adaptation. In response, parameter-efficient fine-tuning methods have emerged [Liu et al., 2022], with Low-Rank Adaptation (LoRA) [Hu et al., 2021] gaining popularity. LoRA introduces trainable low-rank matrices to adapt pre-trained weights, significantly reducing the number of trainable parameters while maintaining competitive performance and offering memory efficiency. While LoRA offers substantial benefits, opportunities remain for **capturing more diverse representations within the limited parameter space**. This is particularly relevant for complex tasks or datasets with limited samples. Additionally, the single low-rank adaptation path in LoRA may not fully exploit the rich information contained in

*equal contribution
†correspondence author

38th Workshop on Fine-Tuning in Machine Learning (NeurIPS 2024).



a) Multi-LoRA b) Share-LoRA c) MPLoRA (ours)

Figure 1: Overview of our motivation and proposed method. a) Multi-LoRA: A group of independent low-rank adapters trained on the same task. b) Share-LoRA: Extends Multi-LoRA by sharing the downsample component (matrix A) across all adapters. c) Multi-Path LoRA (MPLoRA): Further refines the approach by sharing the full downsample matrix, introducing orthogonality constraints among a batch of mini-LoRAs, and concatenating their outputs.

task domain. Addressing these challenges while preserving LoRA's efficiency benefits remains an active area of research, as the field continues to seek optimal solutions for adapting LLMs to specific domains and tasks efficiently.

Ensemble learning [Dong et al., 2020], which combines multiple models to create a stronger predictor, has been widely adopted in machine learning to enhance performance and reduce overfitting. The concept of combining LoRA with ensemble learning is intuitive, as each LoRA can naturally be treated as a separate learner [Huang et al., 2023]. However, most previous works applying ensembled LoRA focus on scenarios like multi-task learning [Dou et al., 2023] or continual learning [Wang et al., 2023], where each LoRA is responsible for acquiring knowledge from different tasks or time periods. The potential of ensemble techniques within a single task, particularly for improving LoRA's representational capacity and performance, remains largely unexplored.

In this work, we explore **ensemble multiple low-rank adapters within the same task domain** to enhance LoRA's performance. We begin by horizontally stacking multiple LoRAs trained on the same dataset, demonstrating that scaling the number of LoRAs can improve performance to some extent. However, we observe that when multiple LoRAs extract knowledge from the same feature space, performance can become unstable. To mitigate this, we introduce an orthogonality loss, encouraging each adapter to learn different aspects of the data. Furthermore, inspired by feature compression techniques, we propose a compress-then-concatenate strategy to aggregate the diverse outputs of multiple adapters. This novel approach, which we call Multi-Path LoRA (MPLoRA), achieves better performance while maintaining the overall parameter count of the original LoRA, effectively enhancing its representational capacity without sacrificing efficiency.The contributions of this work are threefold:

1. We introduce Multi-Path LoRA (MPLoRA), a novel parameter-efficient fine-tuning method that enhances LoRA's capabilities without increasing parameter count. By incorporating multiple orthogonal learning paths and feature compression, MPLoRA enables more effective task-specific adaptations while maintaining computational efficiency.
2. We provide empirical evidence of MPLoRA's superiority over existing methods. Our results demonstrate improved performance across various natural language processing scenarios, offering quantitative support for the practical benefits of our approach.
3. We offer insights into the behavior of decomposed low-rank adaptations and the role of orthogonality in representation learning. Our analysis reveals how multiple low-rank paths interact within the same task domain, contributing to a deeper understanding of efficient parameter adaptation in machine learning.

2 Preliminaries and Motivation

In this section, we introduce the fundamentals of Low-Rank Adaptation and explore its potential enhancements. We first examine the effects of increasing the number of LoRA modules for a single

task. Then, we present a novel approach that shares feature space across multiple adaptation paths, aiming to improve efficiency while preserving multi-path benefits.

2.1 Low Rank Adaptation

Low Rank Adaptation, introduced by [Hu et al., 2021], is an efficient method for adapting pre-trained models. The core idea of LoRA is to represent weight updates using low-rank decomposition. This approach significantly reduces the number of trainable parameters while maintaining model performance. The key mechanism of LoRA can be expressed mathematically as:

$$h = Wx + \Delta Wx = Wx + BAx \tag{1}$$

In this equation, $W, \Delta W \in \mathcal{R}^{d\times d}$ represents the original pre-trained weight and update weight respectively, while $B \in \mathcal{R}^{d\times r}$ and $A \in \mathcal{R}^{r\times d}$ are low-rank matrices that mimic the update procedure. The input vector is denoted by $x \in \mathcal{R}^d$, and the rank r satisfy $r \ll d$, ensuring a low-rank approximation of the weight update. Matrix A is randomly initialized using a Gaussian distribution, while B is set to a zero matrix. This procedure ensures that the initial update (BA) is zero, preserving the pre-trained model's initial behavior. As training progresses, W remain frozen, while matrices A and B are updated through backpropagation, allowing the model to adapt to new tasks efficiently.

2.2 Learning Multiple LoRA Adapters for the Same Task

While LoRA has shown remarkable effectiveness and efficiency in adapting LLMs, individual LoRA adapters may struggle to fully capture the complex representation space necessary for optimal task performance. The low-rank nature of these adapters[Hu et al., 2021], although computationally efficient, potentially limits their ability to represent the complete spectrum of task-specific knowledge. Drawing inspiration from ensemble learning, we propose training multiple low-rank adapters (Multi-LoRA) on the same task data (Fig. 1a). This approach aims to capture and focus on diverse aspects of the task, with each LoRA serving as a unique *path* for knowledge acquisition. The structure of this Multi-LoRA approach can be mathematically described as:

$$h = Wx + \sum_{i=1}^{n} B_i A_i x \tag{2}$$

where $n \in \mathcal{R}$ is the number of LoRA, and A_i, B_i is the component of the i-th LoRA. We can see from Table 1 that this brings consistent improvement compared with vanilla LoRA baseline over four datasets from GLUE benchmark [Wang et al., 2018]. Each low-rank adapter can be viewed as a unique path for extracting distinct representative knowledge from the dataset. As the number n increases, the model's capacity expands, naturally leading to enhanced performance.

2.3 Improve Efficiency with Shared Downsample Matrix

While the concept of Multi-LoRA offers promising potential for performance improvement, it inevitably raises concerns about increased parameter count and computational requirements. To address these concerns, we introduce Share-LoRA (Fig. 1b), a variant of Multi-LoRA designed to enhance parameter efficiency while still leveraging multiple adaptation paths. Specifically, Share-LoRA achieves this balance by sharing the downsample matrix (the A matrix) across all paths, while allowing the B matrices to remain path-specific. This approach draws inspiration from recent advancements in efficient attention mechanisms, such as grouped-query attention (GQA) [Ainslie et al., 2023], which have demonstrated that sharing feature space can lead to efficiency gains without significant performance degradation.

Formally, Share-LoRA can be expressed as:

$$h = Wx + \sum_{i=1}^{n} B_i A x \tag{3}$$

As evident from Table 1, although Share-LoRA experiences a slight performance decrease compared to Multi-LoRA, it still maintains competitive performance across most datasets and on average with

Table 1: Results of motivation experiments on four natural language processing tasks. Higher is better for all metrics. We also report number of LoRA paths (#Num) and number of trainable parameters (#Params). CE and Orth stand for cross-entropy and orthogonality loss respectively. Bold indicates the best results for each metric; underlined values represent the second-best results.

Model	#Num	#Param	Loss	MRPC	COLA	SST-2	STS-B	Avg.
LoRA	1	295k	CE	87.09	52.07	94.61	90.12	80.97
Multi-LoRA	2	589k	CE	87.91	53.53	94.80	90.45	81.67
Multi-LoRA	3	884k	CE	88.40	54.48	<u>95.03</u>	90.30	82.05
Multi-LoRA	4	1179k	CE	88.56	**55.04**	94.91	<u>90.65</u>	<u>82.29</u>
Share-LoRA	2	442k	CE	87.82	52.33	94.95	90.45	81.39
Share-LoRA	3	589k	CE	88.40	54.26	94.68	90.48	81.96
Share-LoRA	4	737k	CE	<u>88.72</u>	54.67	94.34	**90.71**	82.11
Share-LoRA$_{orth}$	2	442k	CE+Orth	86.76	53.50	94.73	90.34	81.33
Share-LoRA$_{orth}$	3	589k	CE+Orth	88.32	54.48	94.92	90.54	82.07
Share-LoRA$_{orth}$	4	737k	CE+Orth	**88.97**	<u>55.03</u>	**95.15**	90.13	**82.32**

much fewer parameters. However, an interesting observation emerges in the SST-2 dataset, where Share-LoRA's performance declines as n increases. This phenomenon suggests potential instability when multiple paths share the same feature space. It highlights the need to further investigate the balance between shared representations and path-specific adaptations in multi-path architectures.

3 Method

Inspired by the performance improvements from scaling up the number of paths, we propose Multi-Path LoRA (MPLoRA), an extensible framework for parameter-efficient fine-tuning (Fig. 1c). First, we introduce orthogonality loss to encourage the model to learn diverse representations from the shared feature space. After that, we propose a compress-then-concatenate strategy to reduce parameter size while maintaining model diversity.

3.1 Adaptation in Orthogonal Subspaces

As previously discussed, each matrix B_i acts as a unique path for extracting knowledge from the feature space generated by matrix A in our Multi-LoRA framework. To encourage these B_i matrices to learn distinct aspects of the feature space, we introduce an orthogonality loss. This loss function promotes diversity among the learned representations by minimizing the similarity between different paths, pushing each path to focus on unique aspects of the input data. Consequently, this enhances the overall effectiveness by maximizing the information captured within the limited parameter space.

Formally, We approximate the parameter update path $\mathcal{U}_i$ for the i-th LoRA as the subspace spanned by the column vectors of B_i:

$$B_i = [b_i^1, b_i^2, \ldots, b_i^r], \quad \mathcal{U}_i = span\{b_i^1, b_i^2, \ldots, b_i^r\} \tag{4}$$

To ensure the orthogonality between different paths, the correspond subspace $\mathcal{U}$ and the subspace $\mathcal{W}$ should satisfy that:

$$< u, w >= 0, \forall u \in \mathcal{U}, \forall w \in \mathcal{W} \tag{5}$$

In the context of our framework, this orthogonality condition translates to $B_i^T B_j = 0$ for $i \neq j$, where B_i and B_j are the matrices spanning these subspaces. To enforce this condition during training, we define the orthogonality loss as:

$$L_{orth}(B_i, B_j) = \sum_{i,j} ||B_i^T B_j||^2 \tag{6}$$

We applied the orthogonality loss to Share-LoRA, with results presented in Table 1. The data shows a notable performance improvement across most scenarios.

Table 2: Performance on natural language processing tasks. Bold indicates best results while underlined shows second-best. Analysis of hyper-parameter setting are provided in Section 5.

Model	#Rank	MRPC	COLA	SST-2	STS-B	QNLI	RTE	Avg.
LoRA	8	87.09	52.07	94.61	90.12	**92.99**	76.17	82.18
AdaLoRA	8	79.90	-	94.72	88.78	92.15	73.76	-
$\text{MELoRA}_{\text{avg}}$	8	86.76	47.18	<u>94.98</u>	89.86	92.58	75.33	81.12
$\text{MELoRA}_{\text{best}}$	8	87.42	49.66	**95.03**	90.22	92.83	75.81	81.83
$\text{MPLoRA}_{\text{avg}}$	8	<u>88.02</u>	<u>52.21</u>	94.76	<u>90.27</u>	92.84	<u>77.54</u>	<u>82.61</u>
$\text{MPLoRA}_{\text{best}}$	8	**88.40**	**52.57**	94.88	**90.32**	<u>92.91</u>	**78.10**	**82.86**

3.2 Multi-path Low Rank Adapter

Drawing inspiration from[Ronneberger et al., 2015] and [Ren et al., 2024], we propose that task-relevant information can be effectively preserved in a slightly reduced dimensional space for each adaptation path. This approach suggests that model performance can be maintained even with smaller feature spaces allocated to individual components in our multi-path system.

Specifically, in our system with n different paths, we down-sample the output length of each LoRA to $\frac{d}{n}$, where d is the original feature dimension, as illustrated in Fig. 1c. Subsequently, we concatenate the outputs of these narrowed LoRAs to restore the original feature length:

$$h = Wx + (concat_{i=1}^{n} B_i^s A)x = Wx + (concat_{i=1}^{n} B_i^s)Ax \tag{7}$$

where $A \in \mathcal{R}^{r \times d}$, $B_i^s \in \mathcal{R}^{r \times \frac{d}{n}}$, $x, h \in \mathcal{R}^d$, and $B = \sum_{i=1}^{n} B_i^s$. For initialization, we adopt the same approach as LoRA: A is initialized with Gaussian distribution, while each B_i^s is set to zero.

After incorporating the orthogonality loss, our learning objective becomes:

$$\sum_{x,y \in D} logp_{\Theta}(y|x) + \lambda \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} L_{orth}(B_i^s, B_j^s) \tag{8}$$

where D represents the training dataset, x and y are input-output pairs from D, Θ denotes the model parameters, and λ is the hyper-parameter that balance the losses.

4 Experimental Setups

4.1 Baselines

1. **LoRA** [Hu et al., 2021] employs low-rank matrix products to learn incremental updates. This approach substantially reduces GPU memory usage during model fine-tuning.
2. **AdaLoRA** [Zhang et al., 2023] introduces adaptive singular value pruning for optimizing matrix ranks. It assigns different ranks to various layers based on singular value magnitudes.
3. **MELoRA** [Ren et al., 2024] stacks multiple mini LoRAs in parallel. It constructs an equivalent block diagonal LoRA matrix by concatenating mini LoRAs along the diagonal.

4.2 Datasets

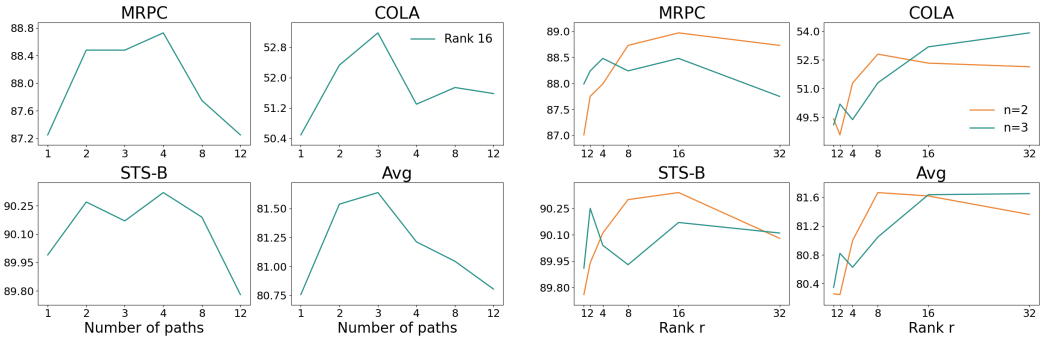
We evaluate the performance on vary tasks in General Language Understanding Evaluation (GLUE) benchmark [Wang et al., 2018], including single-sentence tasks CoLA and SST-2, similarity and paraphrasing tasks MRPC and STS-B, and natural language inference tasks RTE and QNLI. Detailed description of the datasets can be found at Supplementary 1.1.

4.3 Implementation Details

In our experiments, we follow settings from previous works and fine-tune the W_Q and W_V parameters, as suggested by [Ren et al., 2024]. We utilize models and datasets from Huggingface [3], conducting

[3]https://huggingface.co/

fine-tuning on NVIDIA V100 GPUs.We reproduce the result of LoRA and AdaLoRA based on peft [4], and MELoRA base on public code[5]. Results reported are averaged over 3 runs with different random seeds. AdaLoRA results on CoLA consistently yielded 0 scores and are thus omitted. We employ RoBERTa-base[Liu, 2019] as our foundational language model. To ensure fair comparisons, we adopt training configurations based on [Ren et al., 2024], with details provided in the Supplementary 1.2. We set the rank r to 8 for LoRA and its variants, including MELoRA, to maintain consistent parameter number across all methods. For our proposed method, we explore values of n from {2,4,8}, reporting both the best and average performance. Performance metrics vary by task: for CoLA, we report Matthew's correlation; for STS-B, Pearson correlation; and accuracy for the remaining tasks. Across all metrics, higher values indicate superior performance.



(a) Performance of MPLoRA with different paths n. When $n = 1$, MPLoRA degrades to LoRA.

(b) Performance of MPLoRA with different rank r and fixed path n on different datasets.

Figure 2: Analysis of performance across different hyper-parameter settings, where "Avg." denotes the mean score on four natural language processing tasks: MRPC, CoLA, STS-B, and SST-2.

5 Results and Analysis

5.1 Performance on GLUE datasets

The results of all methods on six GLUE datasets are shown in Table 2. MPLoRA demonstrates superior performance compared to other baseline methods across the majority of datasets under equivalent parameter configurations. Notably, significant improvements are observed in datasets with limited training samples, such as MRPC, RTE, and CoLA. We attribute this enhanced performance to MPLoRA's capacity to promote diverse representations among its LoRA components, each capturing unique data aspects. The concatenation of these diverse outputs likely contributes to improved model robustness and generalization. Additionally, MPLoRA's competitive performance on other datasets demonstrates its stability and reliability across various experimental settings.

5.2 Analysis of the Number of Paths

The parameter n denotes the number of paths in the feature space. As shown in Fig. 2a, model performance initially improves with increasing n across all datasets, but eventually declines. This trend can be explained by two competing factors: a larger n allows the model to capture more diverse features, while simultaneously reducing the feature length $\frac{d}{n}$, potentially leading to information loss. Our empirical results suggest that n values of 3 or 4 strike an optimal balance between these effects.

5.3 Impact of Rank on Model Performance

To assess the impact of the rank r on MPLoRA's performance, we experimented with four tasks under two n settings. Fig. 2b illustrates the results for different r values. When r is extremely small, performance suffers due to the limited information in the thin feature space. Forcing the model to

[4]https://huggingface.co/docs/peft/index

[5]https://github.com/ChasonShi/MELoRA

learn from different aspects with such limited information may lead to inconsistent or unreliable feature representations. As r increases, performance improves, reflecting the model's enhanced capacity to capture relevant features. However, larger r values do not necessarily yield proportional gains. We note minimal performance differences between $r = 8$ and $r = 32$, suggesting a low-rank nature of the model. To balance efficiency and effectiveness, we recommend $r = 8$ or $r = 16$ as optimal choices.

5.4 Effectiveness of Orthogonality Loss

We evaluated the impact of the hyper-parameter λ across various datasets under different settings, with results presented in Table 3. Since the orthogonality loss serves as an auxiliary to the main loss, excessively large λ may lead to distraction from the primary task. Based on our empirical findings, we recommend $\lambda = 0.1$ as a suitable value across all experimental configurations.

Table 3: Impact of λ on model performance. Scores are averaged across MRPC, CoLA, and STS-B datasets, with rank $r = 8$ and $n \in \{2, 4, 8\}$.

λ	0	0.05	0.1	0.15	0.2	0.3	0.5	0.6	0.9	5	10
Score	76.14	76.71	77.00	76.78	76.72	76.60	76.61	76.60	76.47	76.42	75.82

6 Related Work

6.1 Parameter Efficient Fine Tuning

Parameter Efficient Fine-Tuning (PEFT) is a key research direction that aims to enhance model performance while minimizing computational resources. Researchers have developed various PEFT strategies, including adapters [Houlsby et al., 2019], prompt learning [Ding et al., 2022], and fine-tuning of model subsets [Ploner and Akbik, 2024]. Among these, Low-Rank Adaptation (LoRA) [Hu et al., 2021] stands out as a particularly effective method for adapting models to new tasks with minimal additional parameters. Building upon LoRA, we propose an efficient architecture that promotes information diversity without increasing the parameter count.

6.2 Ensemble learning

Ensemble learning [Ganaie et al., 2022], which leverages multiple weaker models to boost performance, has gained widespread adoption. This approach encompasses various strategies, including but not limited to weighing model contributions, combining diverse outputs, and selectively utilizing different models based on input characteristics. Among ensemble methods, Mix-of-Experts (MoE) [Shazeer et al., 2016] stands out by integrating predictions from multiple specialized sub-models. Ensemble learning has proven effective across various domains, notably in natural language processing [Shen et al., 2019] and computer vision [Riquelme et al., 2021]. Recently, [Jiang et al., 2023] introduced LLM-Blender, a framework for ensembling multiple open-source LLMs. This approach, which combines pairwise ranking for candidate selection with generative fusion for output refinement, demonstrating the potential of ensemble techniques in advancing LLM capabilities.

6.3 Multi-LoRA Architecture

Multi-LoRA Architecture has emerged as a promising direction in enhancing model performance. ReLoRA [Lialin et al.] introduces a merge-and-reinit procedure, periodically integrating LoRA modules into the LLM and reinitializing them during fine-tuning, effectively stacking multiple LoRA modules. LoRAMoE [Dou et al., 2023] extends this concept by incorporating an MoE-style plugin and a Localize Balancing Constraint, addressing world knowledge forgetting while improving multi-task learning. MELoRA [Ren et al., 2024] further advances the Multi-LoRA approach by decomposing LoRA modules into smaller mini LoRAs and stacking them in parallel, aiming to enhance both efficiency and effectiveness. These developments demonstrate the versatility and potential of Multi-LoRA architectures in various learning scenarios.

7 Conclusion and Future Work

In this paper, we introduced MPLoRA (Multi-Path Low-Rank Adaptation), a novel parameter-efficient fine-tuning method that enhances LoRA through orthogonal multi-path learning and matrix decomposition. Our experiments demonstrated MPLoRA's superior performance over existing methods, particularly on datasets with limited samples, while maintaining parameter efficiency.

Though MPLoRA introduces additional hyper-parameters that require tuning, we provided empirical guidelines to address this challenge. Future work could explore auto-tuning methods, cross-task generalization, theoretical analyses of decomposed low-rank adaptations, and extensions to larger datasets and other domains like vision-language tasks.

References

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. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 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, May 2022.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14:241–258, 2020.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*, 2023.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, et al. Loramoe: Revolutionizing mixture of experts for maintaining world knowledge in language model alignment. *arXiv preprint arXiv:2312.09979*, 4(7), 2023.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, November 2018.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, 2023.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

Pengjie Ren, Chengshun Shi, Shiguang Wu, Mengqi Zhang, Zhaochun Ren, Maarten Rijke, Zhumin Chen, and Jiahuan Pei. Melora: Mini-ensemble low-rank adapters for parameter-efficient fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3052–3064, 2024.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*. Openreview, 2023.

Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. Openprompt: An open-source framework for prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 105–113, 2022.

Max Ploner and Alan Akbik. Parameter-efficient fine-tuning: Is there an optimal subset of parameters to tune? In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1743–1759, 2024.

Mudasir A Ganaie, Minghui Hu, Ashwani Kumar Malik, Muhammad Tanveer, and Ponnuthurai N Suganthan. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, 2022.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2016.

Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. Mixture models for diverse machine translation: Tricks of the trade. In *International conference on machine learning*, pages 5719–5728. PMLR, 2019.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, 2023.

Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. Relora: High-rank training through low-rank updates. In *Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ NeurIPS 2023)*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, October 2013.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*, 2005.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, August 2017.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, November 2016.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer, 2005.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second pascal recognising textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*, volume 1. Citeseer, 2006.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9, 2007.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth pascal recognizing textual entailment challenge. *TAC*, 7(8):1, 2009.

Supplementary Material

1.1 Details of the Evaluated Tasks

Single-sentence tasks:

- The SST-2 (Stanford Sentiment Treebank) [Socher et al., 2013] consists of sentences extracted from movie reviews, accompanied by human-annotated sentiment labels. The objective is to accurately predict the sentiment expressed within each given sentence.
- The CoLA (Corpus of Linguistic Acceptability) dataset [Warstadt et al., 2019] consists of judgments on English sentence acceptability, drawn from linguistic theory literature in books and academic journals. Its primary objective is to determine whether a given English sentence is grammatically sound.

Similarity and Paraphrase Tasks:

- The MRPC (Microsoft Research Paraphrase Corpus) dataset [Dolan and Brockett, 2005] comprises pairs of sentences automatically extracted from online news sources. Its primary purpose is to determine whether the sentences in each pair convey the same semantic meaning.
- The STS-B (Semantic Textual Similarity Benchmark) [Cer et al., 2017] comprises pairs of sentences extracted from various sources including news headlines, video and image captions, and natural language inference datasets. Human annotators have assigned each pair a similarity score on a scale of 1 to 5. The objective is to accurately predict these human-assigned similarity scores.

Inference Tasks:

- The QNLI (Question-answering NLI) dataset [Rajpurkar et al., 2016] is derived from a question-answering corpus, featuring pairs of questions and paragraphs. For GLUE, this dataset is adapted into a sentence pair classification task, where each question is paired with every sentence from its associated context. The objective is to identify whether a given context sentence provides the answer to its paired question.
- The RTE (Recognizing Textual Entailment) datasets originate from a sequence of challenges focused on textual entailment [Dagan et al., 2005] [Bar-Haim et al., 2006] [Giampiccolo et al., 2007] [Bentivogli et al., 2009]. This task requires determining whether a given premise logically implies the associated hypothesis.

1.2 Hyper-parameter Setting

The detailed hyper-parameter settings for evaluation datasets are listed in Table 4.

Table 4: Hyper-Parameters for Different datasets

Hyper-Parameter	**SST-2**	**MRPC**	**CoLA**	**QNLI**	**RTE**	**STS-B**
Learning Rate η	5e-4	4e-4	4e-4	4e-4	4e-4	4e-4
Batch Size	64	64	64	64	64	64
Number of Epochs	60	30	80	25	80	40
Weight Decay β	0.1	0.1	0.1	0.1	0.1	0.1
Max Sequence Length	256	256	256	256	512	256
Start Steps K	400	10	100	800	200	200
Update Ratio λ	0.5	0.5	0.5	0.5	0.5	0.5
Rank r	8	8	8	8	8	8
Alpha α	16	16	16	16	16	16
LR Scheduler	Linear	Linear	Linear	Linear	Linear	Linear
Warmup Ratio	0.06	0.06	0.06	0.06	0.06	0.06
Evaluation Metrics	Accuracy	Accuracy	Matthews Corr.	Accuracy	Accuracy	Pearson Corr.