InterLoRA: An Adaptive LoRA Structure Based on The Mechanistic Interpretability of Transformer

Jihao Gu [1 2] **Zelin Wang** [1 2] **Yibo Zhang** [1 2] **Zhisong Bie** [1 2]

Abstract

With the escalating costs associated with fine-tuning large pre-trained models, the significance of parameter-efficient fine-tuning (PEFT) methods has become increasingly evident. Among these methods, we focus on LoRA, which introduces parallel trainable parameters in the multi-head attention component and has demonstrated promising results. However, previous research may have overlooked the mechanistic interpretability of the transformer architecture, especially since PEFT methods are built upon this framework. Drawing on this insight, we propose **InterLoRA**, which integrates LoRA with feature adaptation mechanism into both the attention layer, considering the varying importance of multiple heads, and the Feed-Forward Network (FFN) layer, acknowledging the memory storage characteristics. Experiments conducted on a variety of complex generation tasks highlight the effectiveness of InterLoRA in jointly fine-tuning both components while efficiently managing parameter memory.

1. Introduction

With an increasing number of large language models, such as GPT-J (Wang & Komatsuzaki, 2021) and LLaMA (Touvron et al., 2023), subsequently appeared and the cost of fine-tuning pre-trained models has become increasingly burdensome with the growing size of the models. Consequently, to address these challenges, Parameter Efficient Fine-Tuning (PEFT) methods become more important (Houlsby et al., 2019; Li & Liang, 2021). These methods typically focus on transformer models, where during training, the pre-trained parameters are frozen, while a small set of parameters is introduced available for fine-tuning. For example, the LoRA (Hu et al., 2021) method fine-tunes by introducing learnable matrices alongside the multi-head attention mechanism. Our work focuses on LoRA, which demonstrates high performance across a wide range of tasks.

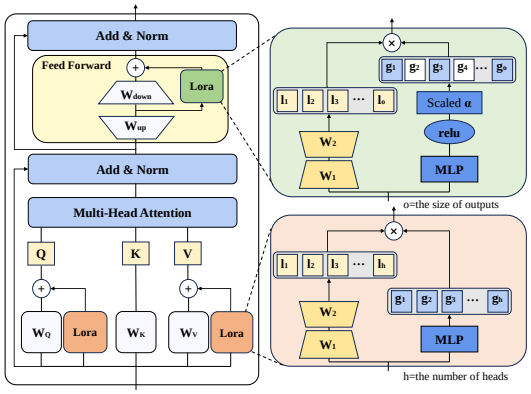

Figure 1. A diagram of the InterLoRA with the transformer structure on the left and our proposed improvement on the right. Here, h corresponds to the number of attention's heads, and o corresponds to the size of the LoRA's output.The white bar for g_i represents zero.

In prior research, the study by He et al. (2022) has shown that LoRA can achieve effective fine-tuning not only in the multi-head attention layers but also in the FFN layers. However, previous works have largely overlooked the differences in transformer mechanistic interpretability between these two components. Specifically, Chefer et al. (2021a;b) have demonstrated that each attention head captures distinct short-term information and exhibits varying levels of importance. Meanwhile, Geva et al. (2020); Dai et al. (2021) have highlighted that the FFN layers, in addition to acquiring more complex information through non-linear transformations, store the majority of the pre-trained model's long-term memory. Based on these observations, we argue that applying different LoRA methods, tailored to their respective interpretability, to both the attention and FFN layers during fine-tuning will better leverage the potential of LoRA techniques.

Building upon the aforementioned insights, we propose a

[1]Beijing University of Posts and Telecommunications, BeiJing, China [2]School of Artificial Intelligence. Correspondence to: Ping Gong <pgong@bupt.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

simple yet effective InterLoRA method that can adaptively combine fine-tuning in both the attention (atten) and FFN sections of the transformer. Our method employs gating units to independently modulate the multiple heads in the attention mechanism and the memorized neurons in the FFN. Specifically, as illustrated in figure 1, for the attention section, we introduce an MLP layer that transforms the input into gate units. These gate units serve as weights corresponding to each multi-head. As for the FFN section, we utilize an MLP to transform the input into the LoRA's corresponding output dimension. Through a ReLU activation function, we selectively filter out neurons that retain pre-trained memory without fine-tuning. The parameter α is employed to balance the filtered results. Subsequently, these results are multiplied with the computed original LoRA values.

We conduct extensive experiments using LLMs with zero-shot learning which demonstrates a significant performance improvement in different situations with our approach on mathematical tasks and commonsense inference tasks, indicating its advantage in leveraging generative capabilities to address complex tasks which is in line with the trend of using large language models. Additionally, empirical evidence supports the effectiveness of our approach in combining the benefits of LoRA fine-tuning in both attention and FFN sections while selectively fine-tuning memory units.

2. Methodology

2.1. LoRA

The transformer consists of two crucial components, namely, multi-head attention and FFN. The formulas for these two sections are as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V \quad (1)$$

$$FFN(x) = ReLU(xW_{up} + b_1)W_{down} + b_2 \quad (2)$$

For LLaMA, the calculation formula for its FFN section is slightly different:

$$FFN_{LLaMA}(x) = (Swish_1(xW_{gate}) \otimes xW_{up})W_{down} \quad (3)$$

In the LoRA method, two matrices, $W_1 \in R^{d_{hidden} \times r}$ and $W_2 \in R^{r \times d_{hidden}}$, are added next to Q and V in the multi-head attention. For example, the calculation formula for Q is as follows:

$$Q = x(W_Q + \lambda W_1 W_2) \quad (4)$$

2.2. InterLoRA

For the multi-head attention section, to dynamically adjust LoRA based on the input from each layer through gate units corresponding to the number of attention heads, we first transform the input into one dimension through a linear layer, followed by another linear layer to obtain the dimension corresponding to the number of heads, denoted as h For each layer and each head, the gate g_{Atten}=[g_1,g_2...g_h] with values in the range (0,1) is computed as follows:

$$g_{Atten} = mean(sigmoid(xW_{Atten}W_h)) \quad (5)$$

Here, x represents the input from the previous layer with a length of m and dimension d, $W_{Atten} \in R^{d_{hidden} \times 1}$ and $W_h \in R^{1 \times d_{head}}$ are learnable parameters, 'mean' denotes taking the average along the sentence dimension, (i.e., averaging for each word in the sentence) to ensure that each word vector corresponds to the same gate at the same head. Subsequently, the calculated result of LoRA is divided into multiple heads (l_{Atten}=[l_1,l_2...l_h]), and each is multiplied with the corresponding g_i The modified LoRA for each head in the attention section is calculated as follows:

$$l'_i = l_i \otimes g_i \quad (6)$$

For the FFN section, previous research has indicated that each parameter in the FFN of a pre-trained model corresponds to the model's memory unit (Dai et al., 2021). Therefore, during fine-tuning, we believe that some parameters need adjustment while others should remain unchanged. To achieve this fine-grained fine-tuning, we transform the input through two linear to match the dimensions of the LoRA output. We apply the ReLU activation function to set some parameters to 0, aiming to fine-tune only the memory parameters that are needed. Subsequently, we introduce a trainable parameter α, to balance the values obtained by ReLU, preventing them from being too small. The final gate $g_{FFN} \in R^{m \times d_{outputs}}$ is calculated as follows:

$$g_{FFN} = \alpha(relu(hW_{FFN}W_o)) \quad (7)$$

Here, h represents the input to the FFN layer after passing through attention, with a length of m and dimension d. $W_{FFN} \in R^{d_{hidden} \times 1}$ and $W_o \in R^{1 \times d_{outputs}}$ are learnable parameters. Afterward, the computed LoRA l_{FFN} is multiplied element-wise with the gate in the output dimension. Through experimental validation (in Section 3.3), we choose to apply LoRA to the downsampling matrix W_{down} of the FFN. The modified matrix formula is as follows:

$$W'_{down} = W_{down} + l_{FFN} \otimes g_{FFN} \quad (8)$$

3. Experiment

3.1. Experimental Setup

In line with the trend of using large language models, we conduct hundreds of experiments across 13 datasets related

LLMs		Params	SVAMP	AQuA	AddSub	MultiA	SingleEQ	GSM8K	Avg.
LLaMA-7b	LoRA	5.24M	58.50	23.53	75.95	**92.73**	**88.24**	24.24	60.53
	Adapter	201.33M	53.50	23.53	74.68	86.36	75.49	20.08	55.61
	prefix	7.86M	42.50	23.53	58.23	60.00	66.67	15.91	44.47
	InterLoRA	**4.78M**	**60.50**	**25.49**	**79.75**	91.82	85.29	**25.38**	**61.37**
LLaMA-13b	LoRA	8.19M	66.00	21.57	82.28	95.45	**89.22**	36.74	65.21
	Adapter	314.57M	55.00	**31.37**	73.42	78.18	69.61	17.05	54.10
	prefix	12.29M	58.00	29.41	72.15	78.18	82.35	22.73	57.14
	InterLoRA	**7.48M**	**69.50**	**31.37**	**84.81**	**96.36**	**89.22**	**38.26**	**68.25**
GPT-J-6b	LoRA	**4.59M**	47.00	5.88	65.82	72.73	76.47	11.36	46.54
	Adapter	117.44M	43.00	13.73	56.96	76.36	64.71	9.85	44.10
	prefix	6.88M	41.50	9.80	**67.09**	75.45	71.57	9.85	45.88
	InterLoRA	4.93M	**48.50**	**23.53**	**67.09**	**80.91**	**77.45**	**13.26**	**51.79**

Table 1. The accuracy results on six mathematical reasoning datasets.(**bold**: the best score)

LLMs		Params	ARC-c	ARC-e	Boolq	WinoG	PIQA	SIQA	OBQA	Avg.
LLaMA-7b	LoRA	5.24M	61.43	77.86	**64.37**	63.22	**75.90**	69.45	71.80	**69.15**
	Adapter	201.33M	46.08	63.34	41.96	53.67	15.23	50.82	50.60	45.96
	prefix	7.86M	49.74	66.20	62.14	50.91	66.21	57.68	51.60	57.78
	InterLoRA	**4.78M**	**61.95**	**79.59**	63.49	**63.46**	71.55	**69.60**	**74.40**	**69.15**
LLaMA-13b	LoRA	8.19M	**68.77**	**83.80**	**68.81**	**68.59**	**80.25**	**72.77**	76.60	**74.37**
	Adapter	314.57M	50.17	65.66	62.17	50.59	68.88	64.33	62.40	60.60
	prefix	12.29M	59.90	77.15	64.46	62.27	74.86	66.33	66.40	67.34
	InterLoRA	**7.48M**	67.15	83.54	68.78	67.88	78.56	71.49	**78.20**	73.66
GPT-J-6b	LoRA	**4.59M**	13.82	16.50	**62.17**	47.51	43.09	33.32	21.80	34.03
	Adapter	117.44M	33.96	49.45	**62.17**	**50.99**	52.01	45.50	35.60	47.10
	prefix	6.88M	15.61	17.13	0.03	0.24	25.73	8.75	5.60	10.44
	InterLoRA	4.93M	**45.48**	**62.96**	61.13	**48.54**	**61.75**	**55.42**	**54.60**	**55.70**

Table 2. The accuracy results on the seven commonsense inference datasets.(**bold**: the best score)

to complex mathematical reasoning and commonsense inference using the understanding capabilities of three large language models through zero-shot methods, mainly referencing the prior work (Hu et al., 2023) with pre-trained parameters from Hugging Face's (Wolf et al., 2020) large language models, including LLaMA-7b/13b (Touvron et al., 2023), and gpt-j-6b (Wang & Komatsuzaki, 2021). The datasets for mathematical reasoning are (1) the SVAMP (Patel et al., 2021), (2) the AQuA (Ling et al., 2017) dataset, (3) the AddSub (Hosseini et al., 2014) dataset, (4) the MultiArith (Roy & Roth, 2016) dataset, (5) the SingleEQ (Koncel-Kedziorski et al., 2015) dataset, and (6) the GSM8K (Cobbe et al., 2021) dataset. The commonsense inference tasks are as follows:(1) the ARC-c and (2) the ARC-e are the Challenge Set and Easy Set of ARC (Clark et al., 2018), (3) the Boolq (Clark et al., 2019), (4) the WinoGrande (Sakaguchi et al., 2021), (5) the PIQA (Bisk et al., 2020), (6) the SIQA (Sap et al., 2019), and (7) the OBQA (Mihaylov et al., 2018)

Since concurrently parallelizing LoRA in both the attention and FFN layers would double the parameter count, we set the LoRA parameter r in our method to 4, half of the baseline, and initialize the learnable parameters α to 10. We compare our method and conduct comparisons with the LoRA (Hu et al., 2021), Adapter (Houlsby et al., 2019) and Prefix (Li & Liang, 2021) methods in our experiments.

3.2. Experimental Results

We present the experimental results on mathematical reasoning and commonsense inference tasks in Tables 1 and 2, respectively. It is evident that our method achieves better performance on almost all datasets. Excluding the poor performance of the GPT-J model using the prefix method in commonsense inference tasks, compared with other methods, our approach can achieve up to a 38% improvement in mathematical reasoning tasks and up to a 50% improvement in commonsense inference tasks. On LLaMA-13b the number of trainable parameters is less than that of the baseline, leading to a slight decrease in commonsense inference datasets' performance. However, this does not affect our method's ability to achieve better results on LLaMA-7b and GPT-J. Although the trainable parameter count for the adapter method is larger than that of our method, our method still exhibits superior performance. The Appendix C.1 provides a low-resource performance of reducing the size of r in LoRA. These experiments collectively demonstrate the effectiveness of our method across various tasks. Our method has fewer parameters in all LLaMA models.

Settings	SVAMP	AQuA	AddSub	MultiA	SingleEQ	GSM8K	Avg.
InterLoRA	**60.50**	25.49	**79.75**	91.82	85.29	25.38	**61.37**
InterLoRA*	54.50	**31.37**	75.95	88.18	**87.25**	25.76	60.50
w/o gate on atten	58.50	13.73	74.68	**93.64**	**87.25**	**29.17**	59.49
w/o gate on FFN	55.50	29.41	74.68	90.00	86.27	26.14	60.33

Table 3. Ablation experiments on the mathematical reasoning dataset, where 'InterLoRA*' represents replacing the ReLU activation function in the method with sigmoid and removing the trainable parameter α.

Settings	ARC-c	ARC-e	Boolq	WinoG	PIQA	SIQA	OBQA	Avg.
InterLoRA	**61.95**	**79.59**	63.49	**63.46**	71.55	**69.60**	**74.40**	**69.15**
InterLoRA*	58.28	78.41	63.27	61.25	**76.06**	66.94	69.20	67.63
w/o gate on atten	59.22	78.32	**64.74**	63.61	74.54	67.35	73.00	68.68
w/o gate on FFN	59.90	78.70	62.97	62.98	75.52	68.42	72.00	68.64

Table 4. Ablation experiments on the commonsense inference dataset. Experiments are consistent with Table 3.

3.3. Analysis of the Role of Each W Matrix

Using LLaMA-7b with similar trainable parameter settings, experiments are conducted on two tasks. We compare the original LoRA method with variations where LoRA is individually applied to the W_{down}, W_{gate}, and W_{up} matrices in FFN, as well as the separate application of these matrices in FFN and attention sections using our joint method. As shown in Figure 2, compared to the original LoRA method, fine-tuning only the FFN part matrices does not lead to significant improvement. In joint experiments, applying LoRA to W_{down} in the FFN demonstrates higher improvement and better performance,and this configuration demonstrates its superiority across various experiments in our study.

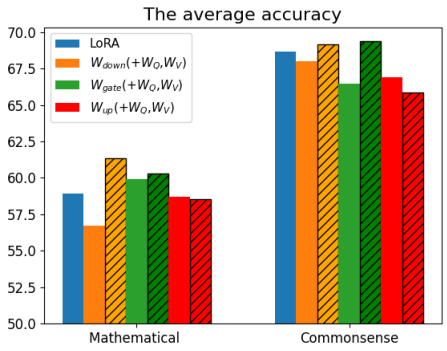

Figure 2. The average accuracy on two tasks with a similar trainable parameter count. On the left side of the same color are results with only individual matrix in FFN, and on the right side, shaded, are results from our joint fine-tuning method applied to both the attention and FFN section

3.4. Ablation Experiments

To verify the effectiveness of each component in our method, we conduct ablation experiments using LLaMA-7b. We separately remove the gate mechanisms in the attention and FFN sections and both of them. Additionally, for a more granular validation of the role of the ReLU activation function in controlling the memory units during FFN fine-tuning, we replace this activation function with the same sigmoid as in the attention section, and remove the trainable parameter α. The experimental results are shown in Tables 3 and 4. It can be seen that simply stacking LoRA in the attention and FFN sections does not achieve satisfactory results. Both components are indispensable, jointly allowing the model to simultaneously excel across multiple tasks and achieve the best overall performance. Although not using the ReLU activation mechanism can still yield good performance in mathematical reasoning tasks, there is a significant decline in performance on commonsense inference tasks. As commonsense inference tasks require more model-specific memory, this also illustrates the superiority of controlling fine-tuning in FFN neurons.

4. Conclusion

In this paper, we have conducted a thorough investigation and optimization of the LoRA method, proposing an enhanced InterLoRA. By concurrently applying LoRA to both the attention and FFN sections of the transformer architecture, we have leveraged the mechanistic interpretability of the model, incorporating distinct feature adaptation mechanisms tailored to each component. Through extensive experiments across various tasks, we have not only validated the effectiveness of our proposed method but also scrutinized the performance of the LoRA method on the weight matrices of the FFN layer and analyzed the roles of different InterLoRA components. Collectively, these experiments demonstrate that InterLoRA significantly extends the capabilities of the original LoRA method, highlighting the potential of PEFT strategies that are grounded in a deep understanding of transformer mechanistic interpretability.

References

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Chefer, H., Gur, S., and Wolf, L. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 397–406, 2021a.

Chefer, H., Gur, S., and Wolf, L. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 782–791, 2021b.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B., and Wei, F. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., and Neubig, G. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=0RDcd5Axok.

Hosseini, M. J., Hajishirzi, H., Etzioni, O., and Kushman, N. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 523–533, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1058. URL https://aclanthology.org/D14-1058.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Hu, Z., Lan, Y., Wang, L., Xu, W., Lim, E.-P., Lee, R. K.-W., Bing, L., and Poria, S. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*, 2023.

Koncel-Kedziorski, R., Hajishirzi, H., Sabharwal, A., Etzioni, O., and Ang, S. D. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597, 2015. doi: 10.1162/tacl_a_00160. URL https://aclanthology.org/Q15-1042.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353.

Ling, W., Yogatama, D., Dyer, C., and Blunsom, P. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1015. URL https://aclanthology.org/P17-1015.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Patel, A., Bhattamishra, S., and Goyal, N. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168.

Roy, S. and Roth, D. Solving general arithmetic word problems. *arXiv preprint arXiv:1608.01413*, 2016.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Sap, M., Rashkin, H., Chen, D., LeBras, R., and Choi, Y. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Wang, B. and Komatsuzaki, A. Gpt-j-6b: A 6 billion parameter autoregressive language model, 2021.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Zhang, Q., Chen, M., Bukharin, A., He, P., Cheng, Y., Chen, W., and Zhao, T. Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

A. Experimental Details

Data Usage: The datasets and hyperparameter settings in this paper are mainly referenced from open-source code (Hu et al., 2023). For the mathematical reasoning tasks, all six datasets are combined by randomly selecting 80% of each, resulting in a total of 3260 data points for training. Testing is then performed on the remaining data for each dataset. For commonsense inference tasks, considring the ressource,15k version of this work(Hu et al., 2023) are used for training, and testing is conducted on the seven datasets mentioned in the main text. During training and testing, a prompt is added to the data: 'Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.'

Hyperparameter Settings: The learning rate for both our method and the original LoRA method is set to 3e-4, and the regularization parameter λ in Eq 4 of the main text is set to 2,and in figure2,3 and 5 the value of r for original LoRA was set to 8 following the reference methods(Hu et al., 2023), while in the main experiment, the value of r for LORA in comparison was set to 10 to demonstrate that the baseline method still could not achieve comparable results even with a higher parameter count. For the prefix method, the learning rate is set to 3e-2, and the length of virtual tokens is set to 30. The bottleneck size for the adapter method is set to 256. To ensure comparability and reproducibility of experiments, the random seed for all experiments is set to 42 and all training epochs are set to 3.

Model Usage: In this paper, we utilize three large models: LLaMA-7b/13b (Touvron et al., 2023), and GPT-J-6b (Wang & Komatsuzaki, 2021). All training and testing experiments are conducted using either a single Nvidia A40 or Nvidia RTX4090.

B. Related Works

With the widespread adoption of large models, methods for fine-tuning entire models for downstream tasks become increasingly expensive. The PEFT methods have gradually played a significant role and are becoming widely applied. The most commonly used methods include Adapter (Houlsby et al., 2019), Prefix (Li & Liang, 2021), and LoRA (Hu et al., 2021), all of which involve freezing pre-trained parameters in the transformer structure and introducing a portion of trainable parameters. In the Adapter method, a trainable module is serially concatenated to both the attention and FFN parts of the model. The Prefix tuning method introduces a certain length of virtual tokens before multi-head attention for fine-tuning. The LoRA method, on the other hand, parallelly adds two trainable matrices to the attention part for fine-tuning in downstream tasks.

Recently, adaLoRA (Zhang et al., 2023) simultaneously placed LoRA in both the attention and FFN sections, dynamically allocating parameter budgets to the weight matrices based on importance scores, and validated its performance on language understanding tasks. QLoRA (Dettmers et al., 2023) fine-tuned quantized models to 4 bits without compromising any performance. Unlike the privoir methods, our commitment lies in effectively integrating LoRA fine-tuning in both attention and FFN parts with the principles of transformer mechanistic interpretability, aiming to enhance the performance of PEFT. Besides, in contrast to the majority of approaches that experiment with smaller models on understanding tasks, we conduct experiments using zero-shot learning on LLMs.

C. Supplementary Experiments

C.1. Analysis under Low Resource Conditions

To investigate the applicability of our model under various low-resource scenarios, we reduce the parameter count using LLaMA-7b and simultaneously reduce the r values for both LoRA and our method by almost 2 times. The experimental results are shown in Table 5, indicating that even in these low resource conditions, our method still performs well.

C.2. Analysis of Fine-Tuning Other Matrices Combination

We also compare the fine-tuning of all W matrices in the FFN part as well as only fine-tuning W_{down}, W_{gate}, and W_{up} matrices using the LoRA method simultaneously with the LLaMA-7b model. The experimental method is the same as described in Section 3.3 of the main text, comparing the LoRA method with the InterLoRA method proposed in this paper under similar parameter conditions. The experimental results are shown in Figure 3. Although other methods have achieved decent performance, the proposed InterLoRA method still achieves the best performance.

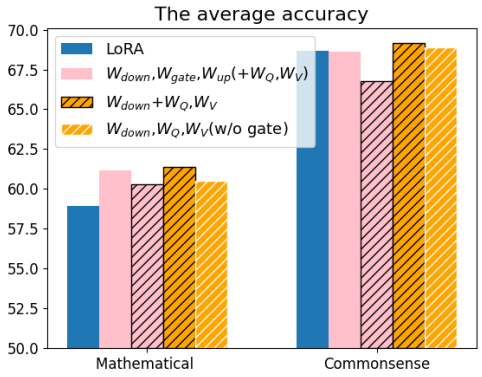

Figure 3. The average accuracy on two tasks with a similar trainable parameter count. On the left side of the same pink color are results with only individual matrix in FFN, and on the right side, shaded, are results from our joint fine-tuning method applied to both the attention and FFN sections

Settings	SVAMP	AQuA	AddSub	MultiA	SingleEQ	GSM8K	Avg.
InterLoRA-	**54.50**	19.61	**78.48**	90.00	83.33	**26.52**	**58.74**
LoRA(r=5)	51.50	**23.53**	73.42	**90.91**	**87.25**	23.48	58.35

Table 5. Comparative experiments with reduced parameter count on the mathematics datasets, where - denotes reducing the rank r by half

D. Scientific Artifacts

The datasets we use include the mathematical reasoning dataset SVAMP (Patel et al., 2021), AQuA (Ling et al., 2017), AddSub (Hosseini et al., 2014), MultiArith (Roy & Roth, 2016), the SingleEQ (Koncel-Kedziorski et al., 2015), GSM8K (Cobbe et al., 2021), and the commonsense inference dataset ARC (Clark et al., 2018), Boolq (Clark et al., 2019), WinoGrande (Sakaguchi et al., 2021), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), and OBQA (Mihaylov et al., 2018). The pre-trained models we utilize are LLaMA-7b/13b (Touvron et al., 2023), and gpt-j-6b (Wang & Komatsuzaki, 2021). All the aforementioned datasets and models are open-source, and our work is solely for scientific research purposes, aligning with their original intent.