Learning Global Controller in Latent Space for Parameter-Efficient Fine-Tuning

Zeqi Tan[1], **Yongliang Shen**[1], **Xiaoxia Cheng**[1], **Chang Zong**[1], **Wenqi Zhang**[1], **Jian Shao**[1], **Weiming Lu**[1,2]*, **Yueting Zhuang**[1]*
[1]School of Computer Science and Technology, Zhejiang University
[2]Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies
{zqtan,syl,luwm,yzhuang}@zju.edu.cn

Abstract

While large language models (LLMs) have showcased remarkable prowess in various natural language processing tasks, their training costs are exorbitant. Consequently, a plethora of parameter-efficient fine-tuning methods have emerged to tailor large models for downstream tasks, including low-rank training. Recent approaches either amalgamate existing fine-tuning methods or dynamically adjust rank allocation. Nonetheless, these methods continue to grapple with issues like local optimization, inability to train with full rank and lack of focus on specific tasks. In this paper, we introduce an innovative parameter-efficient method for exploring optimal solutions within latent space. More specifically, we introduce a set of latent units designed to iteratively extract input representations from LLMs, continuously refining informative features that enhance downstream task performance. Due to the small and independent nature of the latent units in relation to input size, this significantly reduces training memory requirements. Additionally, we employ an asymmetric attention mechanism to facilitate bidirectional interaction between latent units and freezed LLM representations, thereby mitigating issues associated with non-full-rank training. Furthermore, we apply distillation over hidden states during the interaction, which guarantees a trimmed number of trainable parameters. Experimental results demonstrate that our approach achieves state-of-the-art performance on a range of natural language understanding, generation and reasoning tasks.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Ouyang et al., 2022; Chowdhery et al., 2022; Zhang et al., 2022; Zeng et al., 2023; Touvron et al., 2023a), exemplified by ChatGPT, have garnered substantial attention within the scholarly and industrial realms owing to their remarkable efficacy in a plethora of natural language processing undertakings. However, full training of LLM is time-consuming and labor-intensive. Besides the high training overhead, maintaining a replica for each task introduces significant storage redundancy.

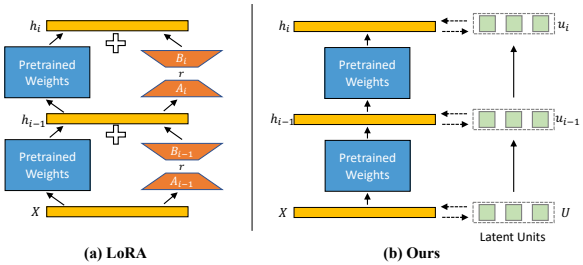

Figure 1: The white plus indicates summing the two hidden state values bitwise, and the dashed arrow represents an optional bidirectional exchange of information. (a) LoRA (Hu et al., 2022) approximates the update of each weight matrix with a pair of A_i and B_i, involving various modules in transformer layers. (b) Instead of this idea of local approximation, our approach treats LLM as a feature extractor and uses a set of latent units to iteratively perform the exchange of information with LLM, further exploiting LLM's power.

To address these issues, researchers either add extra neural modules (Rebuffi et al., 2017; Houlsby et al., 2019; Pfeiffer et al., 2020) or model incremental updates (Zaken et al., 2021; Guo et al., 2021; Hu et al., 2022). More recently, researchers either merge existing fine-tuning methods (He et al., 2022a; Wang et al., 2023a) or dynamically adjust rank allocation (Zhang et al., 2023; Ding et al., 2023a). He et al. (2022a) presents a unified framework and enables an efficient combination of existing fine-tuning methods. Ding et al. (2023a) dynamically allocates the parameter budget among weight matrices. However, these methods still have some flaws. **First**, local optimization is a notorious problem. As shown in Figure 1 (a), LoRA (Hu et al., 2022) approximates the update of each weight matrix with a pair of A_i and B_i. These

* Corresponding author

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4044–4055

low rank-based methods use numerous pairs of low rank matrices to fit local parameters and neglect global control, resulting in suboptimal performance. Zhang et al. (2023); Ding et al. (2023a) remains with this problem even though it makes the unalterable rank in LoRA to be adaptive. **Second**, these methods mostly suffer from inability to train with full rank. He et al. (2022a) finds that this problem manifests itself even more severely in FFNs, and therefore assigns more parameters to FFNs than to attention modules to alleviate this problem. However, as in previous approaches (Pfeiffer et al., 2020; Guo et al., 2021), they still takes a bitwise summing approach to varying the output probabilities, limiting the expressive power of the model. **Third**, these methods focus only on approximate updates to LLMs and do not adequately consider task-specific relevant features (Wang et al., 2023b). Modeling the topics or labels of specific tasks will result in performance gains.

To address these issues, we propose to learn a global controller (GloC) for parameter-efficient training in latent space, in which we use a set of latent units to iteratively distill information features from LLM. As shown in Figure 1 (b), we treats LLM as a feature extractor and uses a set of latent units to perform the exchange of information with LLM. By the nature of the small and independent nature of the latent units relative to the size of the input, this greatly reduces the requirement for training memory. We consider this set of latent units as a global controller that runs through all layers of the large language model, fully exploiting the capabilities of LLM. In addition, we employ asymmetric attentional mechanisms to facilitate bidirectional interactions between latent units and freezed representations, hence mitigating the problems associated with non-full-rank training. Further, we apply a distillation technique to the hidden states during the interaction to compress the hidden state size to a very small scale, which ensures fewer trainable parameters. We also find in our experiments that these latent units learn task-specific relevant features that show strong statistical correlation with task labels. Our main contributions are as follows:

- We consider parameter efficient fine-tuning from a novel perspective that learns a global controller to interact with LLMs in an informative manner. Based on a small set of latent units, we steer the large language model from a global angle, seeking optimal performance.
- We design the asymmetric attention mechanism and distillation compression module during the information exchange to reduce the training memory while mitigating the problem of non-full-rank training.
- Extensive experiments on a range of natural language understanding, generation, and reasoning tasks show that our model reaches the state-of-the-art and significantly outperforms a series of robust baselines. The experimental results also indicate that these latent units model task-specific features.

2 Related Work

2.1 Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) is a set of methods that optimize only a small fraction of the parameters, keeping the backbone model frozen to adapt to downstream subtasks. Mainstream approaches either add external neural modules (Houlsby et al., 2019; Li and Liang, 2021; Lester et al., 2021) or model incremental updates (Zaken et al., 2021; Guo et al., 2021; Hu et al., 2022). Specifically, the methods for adding extra modules include Adapter (Houlsby et al., 2019; Rebuffi et al., 2017; Pfeiffer et al., 2020), Prefix (Li and Liang, 2021) and Prompt Tuning (Lester et al., 2021). Adapter inserts small neural modules called adapters between layers of the backbone model, while Prefix and Prompt Tuning appends additional trainable prefix tokens to the input or hidden layers, similar work also includes P-tuning (Liu et al., 2021). Another mainstream of approaches model incremental updates of pre-training weights without modifying the model structure. (Zaken et al., 2021) only fine-tunes bias vectors in the backbone model, and diff-pruning (Guo et al., 2021) learns a sparse parameter update vector. LoRA (Hu et al., 2022) approximates the update of each weight matrix with a pair of low-rank matrices.

Recent work either customize existing fine-tuning methods (He et al., 2022a; Wang et al., 2023a) or dynamically adjust rank allocation (Zhang et al., 2023; Ding et al., 2023a). He et al. (2022a) presents an efficient combination of existing fine-tuning methods, and Wang et al. (2023a) utilizes low-rank techniques to highly parameterize skills in the multi-task. Zhang et al. (2023); Ding et al. (2023a) both dynamically allocates the parameter budget among weight matrices. Zhang et al.

(2023) prunes the singular values of unimportant updates, while Ding et al. (2023a) use a gate unit to controll the cardinality of rank. However, they all suffer from the problem of local optimization and the inability to train with full rank.

2.2 Controller View for PEFT

Yang and Liu (2022) proposes to explain prefix tuning from a controller perspective. Ding et al. (2023b) extends the controller perspective to a broader set of PEFT approaches. They argue that the essence of PEFT lies in the regularized layered hidden state transformation process. The proposed global controller is inspired by Lee et al. (2019); Jaegle et al. (2021), which are designed to address high-dimensional multimodal inputs. Unlike their attempts to compress inputs of tens of thousands of dimensions (e.g., pixels) into lower units for probabilistic generation, our approach establishes bi-directional channels for information exchange and distilling hidden states.

3 Method

3.1 Preliminaries

Transformer-based Models A typical transformer model is composed of a stack of L transformer (Vaswani et al., 2017) layers, and the modeling process mainly involves the multi-head attention mechanism. For simplicity, we denote the attention as:

$$\text{Attention}\left(Q, K, V\right) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{1}$$

where Q, K, V are the query, key and value matrix respectively, and the $1/\sqrt{d_k}$ is the scaling factor. Given the input token embeddings $X \in \mathbb{R}^{n\times d}$, n is length of input token sequence and d is hidden size, the multi-head self-attention (MHA) computes the output on N_h head and concatenates them:

$$\begin{aligned}\text{MHA}(X) &= \text{Concat}\left(\text{head}_1, \cdots, \text{head}_\text{h}\right)\boldsymbol{W}_o, \\ \text{head}_\text{i} &= \text{Attn}\left(E\boldsymbol{W}_q^{(i)}, E\boldsymbol{W}_k^{(i)}, E\boldsymbol{W}_v^{(i)}\right),\end{aligned} \tag{2}$$

where $\boldsymbol{W}_o \in \mathbb{R}^{d\times d}$, $\boldsymbol{W}_q^{(i)}, \boldsymbol{W}_k^{(i)}, \boldsymbol{W}_v^{(i)} \in \mathbb{R}^{d\times d_h}$, d_h is typically set to d/N_h.

PEFT with Transformer In order to provide a comprehensive understanding of the differences between our approach and the previous series of approaches, we do a careful recap here. As in Figure 2, the different peft methods are embedded in

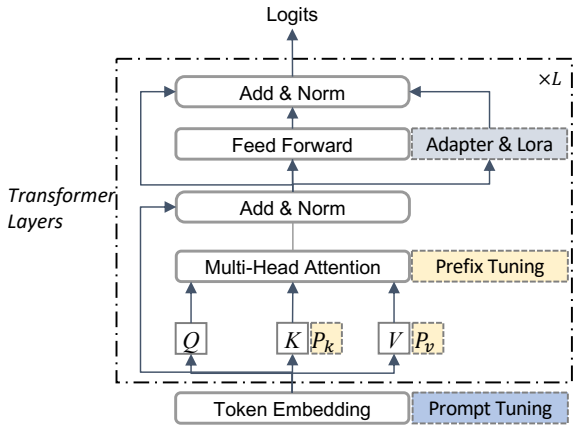

Figure 2: The location of various peft methods in a transformer layer. For simplicity, we only show LoRA approximation to FFN matrices, which can be extended to arbitrary matrices. Similarly, we illustrate parallel adapters, while some earlier adapters are sequential.

different modules of the transformer. It is worth noting that LoRA (Hu et al., 2022) can be used to approximate arbitrary matrices, including the matrices in attention and FFN, although of course approximation of more matrices will result in more parametric quantities. Prefix (Li and Liang, 2021) and Prompt Tuning (Lester et al., 2021) appends additional trainable prefix tokens to the input or hidden attention layers, similar work also includes P-tuning (Liu et al., 2021). Adapter inserts small neural modules called adapters between layers of the backbone model. Houlsby et al. (2019) places two adapters sequentially within one layer of the transformer, one after the multi-head attention and one after the FFN sub-layer, while He et al. (2022a) incorporate extra adapter modules in parallel as in Figure 2. Unlike these approaches that add additional small modules to the submodules of the transformer layers or approximate the update of the local matrices, our proposed approach stands outside of the backbone's transformer layers and delivers the flow of information with a global perspective, as shown on the left side of Figure 3. Moreover, our approach can be summarized as prefix fine-tuning with low-rank attention matrices. We will elaborate on this in the following section.

Latent Units The latent arrays in our approach can be traced back to (Gu et al., 2018; Carion et al., 2020). Carion et al. (2020) refers to them as learnable queries for the set multi-objective detection. Jaegle et al. (2021) uses these latent arrays to compress high-dimensional multimodal inputs

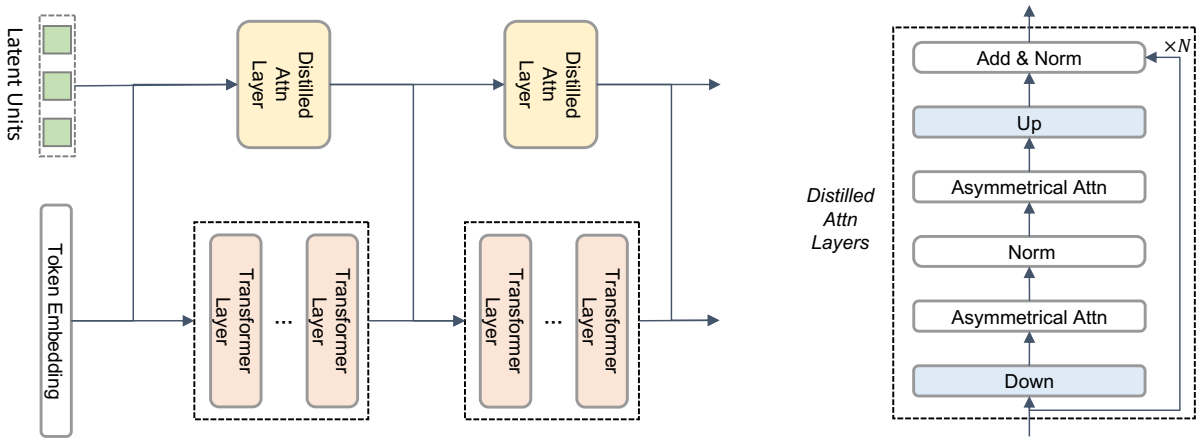

Figure 3: The overall architecture of the proposed Global Controller. On the left, we use a set of latent units as the global controller that runs through all layers of the backbone model to steer the capabilities of LLMs from a global perspective. These latent units iteratively distill information features from the LLM and update the hidden state of the LLM using the distilled attention mechanism. On the right, we show the various modules of the distilled attention mechanism. We first map the hidden state dimensions to lower dimensions, after which we execute the asymmetric attention mechanism on the lower dimensions to facilitate bidirectional interactions between latent units and freezed representations, reducing the training memory while mitigating the problem of non-full-rank training.

and refers to them as latent units. Recently, Wang et al. (2023b) uses these latent variables in context learning, which is utilized to model latent topics or concepts as:

$$P\left(\boldsymbol{x}_{1:T}\right)=\int_{\Theta} P\left(\boldsymbol{x}_{1:T} \mid \boldsymbol{\theta}\right) P(\boldsymbol{\theta}) d \boldsymbol{\theta}, \quad (3)$$

Where $\boldsymbol{\theta} \in \Theta$ represents a latent high dimensional topic or concept variable, Θ is the space of the topic or concept variable, and $\boldsymbol{x}_{1:T}$ refers to the input token embedding.

3.2 Overview

As shown in Figure 3, we treats LLM as a feature extractor and uses a set of latent units $U = \mathbb{R}^{M\times d}$ to perform the exchange of information with LLM. The number of the units M is pre-specified. By the nature of the small and independent nature of the latent units relative to the size of the input, this greatly reduces the requirement for training memory. Formally, for L transformer layers, we use a control factor s for chunking, where s can be divisible by L. After that, for each transformer chunk, we will use a distilled attention mechanism layer to complete the information interaction between latent units and hidden states, where the number of transformer chunks, i.e., the number of distilled attention mechanism layers, is $N = L/s$. When $N = L$, it means that we exchange information at each transformer layer, but this leads to an increase in computation and training memory. We denote the latent units U transformed by the i-th distilled attention layer as $u_i \in \mathbb{R}^{M\times d}$, and similarly, the input embedding X transformed by the i-th transformer block as $h_i \in \mathbb{R}^{n\times d}$. Afterwards, we employ asymmetric attentional mechanisms to facilitate bidirectional interactions between latent units u_i and freezed representations of the backbone model h_i. Since we replace simple summation with bidirectional attentional interactions, we alleviate the problem of training with non-full-rank training. Further, we apply a distillation technique to the hidden states during the interaction to compress the hidden state size to a very small scale, which ensures fewer trainable parameters.

It is worth mentioning that, our approach can be framed as prefix fine-tuning with low-rank matrices constrained by masking. Ding et al. (2023a) mentioned the instability issue in prefix training, which we attribute to the strong coupling between the prefix units and backbone models without additional mapping matrices and masking, resulting in a large difference in the initial variable space. Compared to prefix-tuning, our method has an additional low-rank mapping space (the former two use the QKV mapping of the backbone model itself), which leads to better training convergence while mitigating expressive bottlenecks.

3.3 Distilled Attention

Based on Jaegle et al. (2021), we design distilled attention mechanisms for parameter-efficient training.

Algorithm 1: Distilled Attention

Input: Latent units u_i, token hidden states h_i, down projection weights D_u, D_h, up projection weights P_u, P_h

begin

$u'_i, h'_i \leftarrow u_i D_u, h_i D_h$; // down proj

$u'_i \leftarrow \mathrm{MHA}(u'_i, h'_i, h'_i)$; // attention

$u'_i \leftarrow \mathrm{LN}(u'_i)$; // layer norm

$h'_i \leftarrow \mathrm{MHA}(h'_i, u'_i, u'_i)$; // attention

$u''_i, h''_i \leftarrow u'_i P_u, h'_i P_h$; // up proj

$h''_i \leftarrow \mathrm{LN}(h''_i)$; // layer norm

$u_i, h_i \leftarrow u_i + u''_i, h_i + h''_i$; // residual

Output: u_i, h_i

Different from Jaegle et al. (2021) which predicts the labeling probability of downstream tasks based on latent arrays, we use the asymmetric attention mechanism to perform the information exchange between the latent units and the backbone model, and ultimately generate the probability distribution based on the hidden state of the backbone model. As shown in Algorithm 1, we design the distillation module to project the latent units u_i and hidden state dimensions h_i to lower dimensions as u'_i, h'_i, thus satisfying the need for parameter-efficient training.

Asymmetrical attention We build our information exchange architecture around the attention mechanism because it is both universally applicable and powerful in practice. The main challenge facing traditional attention is that the complexity of Q-K-V self-attention is quadratic in the number of input dimensions, while the length n of the input sequence is usually very large. Here, we apply attention directly to M latent units by introducing asymmetry in the attention operation. The resulting attention operation has complexity $\mathcal{O}(Mn)$. Since the number of latent cells is much smaller than the length of the input sequence ($M \ll n$, e.g., $M = 8$ when $n = 512$), this greatly reduces the computational complexity. In addition, since we use the bidirectional attentional interaction mechanism instead of simple summation in the traditional approach, this helps alleviate the problem of non-full-rank training (He et al., 2022a).

Projection It is worth noting that the inclusion of the Q-K-A mapping matrices in the traditional MHA module introduces a large number of parametric quantities, which is not desired in parameter-efficient fine-tuning methods. In order to reduce the introduction of such parameters, we draw on the technique of distillation (Zhao et al., 2019) that projects the latent units u_i and hidden state dimensions h_i to lower dimensions as u'_i, h'_i. Formally, we denote the down and up projection matrices as $D_u, D_h \in \mathbb{R}^{d \times d'}$ and $P_u, P_h \in \mathbb{R}^{d' \times d}$. As shown in Algorithm 1, we execute the asymmetric attention mechanism in lower dimensions ($d' \ll d$, e.g., $d' = 8$ when $d = 768$), which further reduces the number of computational parameters.

Dataset	#Train	#Valid	#Test	Metric
CoLA	8.5k	1,043	1,063	Mcc
SST-2	67k	872	1.8k	Acc
MRPC	3.7k	408	1.7k	Acc
QQP	364k	40.4k	391k	Acc/F1
STS-B	5.7k	1.5k	1.4k	Corr
MNLI	393k	9.8k/9.8k	9.8k/9.8k	Acc(m/mm)
QNLI	105k	5.5k	5.5k	Acc
RTE	2.5k	277	3k	Acc

Table 1: Dataset statistics and metric in GLUE benchmark. "Mcc", "Acc", "F1" and "Corr" represent matthews correlation coefficient, accuracy, the F1 score and pearson correlation coefficient respectively.

4 Experiments

To demonstrate the effectiveness of the proposed global controller method (GloC), we conduct extensive experiments on a range of natural language understanding, generation, and reasoning tasks.

4.1 Datasets

For evaluation on natural language understanding and generation tasks, we adopt the GLUE benchmark (Wang et al.), including CoLA (Warstadt et al., 2019), SST-2 (Socher et al., 2013), MRPC (Dolan and Brockett, 2005), QQP (Wang et al.), STS-B (Wang et al.), MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016) and RTE (Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009) Table 1 shows the detailed dataset statistics and the evaluation metric. Additionally, for the reasoning task, we take six common-sense reasoning datasets, including BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), etc. BoolQ is a question-answering dataset for yes/no questions containing 15942 examples. These problems occur and arise in unprompted and unconstrained environments. PIQA consists of questions with two solutions requiring physical commonsense to an-

Method	#Params	CoLA	SST-2	MRPC	QQP	STS-B	MNLI	QNLI	RTE
Fine-Tune	184M	69.21	95.64	89.22	92.05/89.31	91.59	89.98/89.95	93.78	82.49
Adapter	1.41M	69.00	95.16	89.90	91.45/88.88	92.21	90.11/90.11	93.79	82.44
Bitfit	0.1M	68.70	94.38	87.16	87.86/84.20	89.71	87.45/87.45	91.90	76.12
LoRA (r=8)	1.33M	69.73	95.57	89.71	91.95/89.26	91.86	90.47/90.46	93.76	85.32
AdaLoRA	1.27M	70.86	95.95	90.22	92.13/88.41	91.39	90.27/90.30	94.28	87.36
SoRA	0.91M	71.48	95.64	91.98	**92.39/89.87**	92.22	90.35/90.38	**94.28**	87.77
GloC	1.33M	**72.35**	**96.16**	**92.31**	92.36/89.54	**92.45**	**90.62/90.45**	94.15	**88.36**

Table 2: Test results of the proposed method and other baselines on the GLUE benchmark. We denote the best result in **bold**. We report mean of 5 runs using different random seeds.

swer. SIQA focuses on reasoning about people's actions and their social implications. The detailed dataset descriptions and data statistics are provided in the Appendix A.1.

4.2 Implementation Details

For fair comparison with prior work, we use DeBERTaV3-base (He et al., 2021) as the backbone model for the GLUE benchmark, Llama (Touvron et al., 2023b) and Llama2 (Touvron and et al., 2023) for the common-sense reasoning datasets. For both versions of a range of sizes of Llama, we adopt 7B size. Commonly, we use the AdamW (Loshchilov and Hutter, 2017) optimizer with a linear warmup-decay learning schedule and a dropout (Srivastava et al., 2014) of 0.1. The latent units are randomly initialized with the normal distribution $\mathcal{N}(0.0, 0.02)$. For hyper-parameters, we set the learning rate to 1e-4 with a batch size of 32. In the main experiments, if not additionally mentioned, we set $M = 8$, $s = 2$ and $d' = 16$. For almost all experiments, we run 5 times using different random seeds and report the average results in order to ensure statistical significance.

4.3 Baselines

In this paper, we compare the proposed GloC with full-parameter fine-tuning and the following robust baseline models: Adapter (Houlsby et al., 2019), BitFit (Zaken et al., 2021), LoRA (Hu et al., 2021), AdaLoRA (Zhang et al., 2023) and SoRA (Ding et al., 2023a). Notably, on the common-sense reasoning task, since full parameter fine-tuning of a large model is unaffordable for a small workshop, we use the results of ChatGPT[1] as an alternative to full fine-tuning. We use GPT-3.5 `text-Davinci-003` for Zero-shot CoT (Kojima et al., 2022) as the baseline.

4.4 Overall Performance

GLUE performance As shown in Table 2, SoRA and AdaLoRA, the state-of-the-art methods on the current ranking, outperform previous methods, including LoRA, on a range of understanding and generation tasks, demonstrating the effectiveness of dynamically adjusting the rank. More evidently, our proposed method achieves state-of-the-art results on six subtasks and comparable results on two others. As an example, GloC outperforms AdaLoRA by 1.49% and 2.09% on CoLA and MRPC, respectively, which demonstrates the effectiveness of our global perspective. We consider the set of latent units as a global controller that runs through all layers of the large language model, fully exploiting the capabilities of LLM.

Reasoning performance On the average F1-measure of the 6 common-sense datasets in Table 7, the result of AdaLoRA (Zhang et al., 2023) improves over the LoRA baseline by +0.4% and +0.7% on different Llama version, while our method further improves by +2.2% and +2.0% compared to AdaLoRA, which speaks volumes about the effectiveness of our approach. In addition, our method outperforms ChatGPT by 3.3% and 3.7% on HellaS and OBQA, respectively, which suggests that parameter-efficient fine-tuning methods still have a lot of potential and room for development when compared to state-of-the-art LLMs. We attribute these performance improvements to that we employ asymmetric attentional mechanisms to facilitate bidirectional interactions between latent units and freezed representations, hence mitigating the problems associated with non-full-rank training. In addition, we believe that the latent units

[1] `https://openai.com/blog/chatgpt/`

Method	# Params	BoolQ	PIQA	SIQA	HellaS	WinoG	OBQA	AVE.
ChatGPT	-	73.1	85.4	68.5	78.5	66.1	74.8	74.4
LLaMA-Adapter	0.77%	67.9	76.4	78.8	69.8	78.9	75.2	74.5
LLaMA-LoRA	0.72%	68.9	80.7	77.4	78.1	76.8	74.8	76.1
LLaMA-AdaLora	0.69%	69.4	80.8	77.8	78.6	77.1	75.4	76.5
LLaMA-GloC	0.72%	72.1	83.6	78.7	81.4	79.2	77.1	78.7
LLaMA2-Adapter	0.77%	68.2	78.1	78.4	71.2	78.1	75.6	74.9
LLaMA2-LoRA	0.72%	70.8	82.4	78.8	78.5	77.4	74.8	77.1
LLaMA2-AdaLora	0.69%	71.2	82.9	78.8	79.6	77.8	76.6	77.8
LLaMA2-GloC	0.72%	**73.8**	**85.3**	**79.1**	**81.8**	**80.4**	**78.5**	**79.8**

Table 3: Results with LLaMA & LLaMA2 on six common-sense reasoning datasets. The best results on each dataset are shown in **bold**. We report mean of 5 runs using different random seeds.

Method	CoLA	STS-B	QNLI
Default	**72.35**	92.45	**94.15**
freezed units	72.24	92.28	94.06
one-way interaction	70.96	90.32	93.57
w/o projection	72.35	**92.62**	94.09
w/ FFN	71.86	90.85	93.64

Table 4: Ablation studies with four different settings of our method on three GLUE datasets.

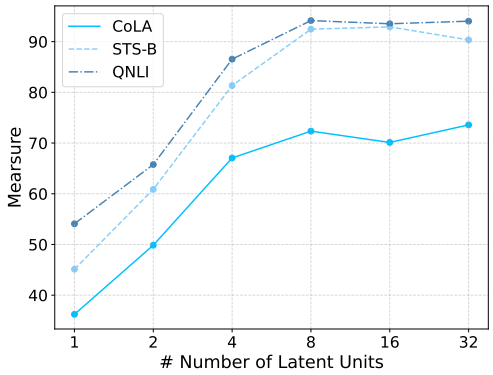

Figure 4: The performance of the model under different number of latent units on the three GLUE datasets.

learn task-specific linguistic information that further improves the performance of the model. We will conduct extensive ablation experiments in subsequent sections to support our points. We also supplement the results of prefix and prompt tuning in the Appendix A.2. More analysis for non-full-rank training can be found in the Appendix A.3.

Another thing to keep in mind is that, we find in practice that GloC has slightly more parametric quantities than LoRA, due to the fact that in addition to the parameters of the projection, the latent cells also take up part of the parametric quantities. We ensure that the number of parameters is basically the same as LoRA by controlling the chunking factor s and the low-dimensional size d' of the projection in our experiments, e.g., when r is 8 in LoRA, we set $s = 2$ and $d' = 16$. We will carefully analyze the impact of these hyperparameters on the model performance.

5 Analysis

5.1 Ablation Study

In this section, we perform extensive ablation experiments to analyze the necessity of each sub-module of the proposed method and the extent of its impact on the performance. We mainly design the following four setup: (1) **freezed units**: we freeze the feature space of latent cells and learn only the projection matrices in the experiments, (2) **one-way interaction**: we set up so that latent units no longer draw information from the backbone model, turning what was originally a bidirectional interaction into a unidirectional one, (3) **w/o projection**: we no longer map the features in low dimensions, keeping the mapping matrices as in the original MHA, which would add much training time, (4) **w/ FFN**: based on (3), we add two additional layers of FFNs after MHA, which leads to a further expansion of the number of parameters.

From Table 4 we observe that **one-way interaction** leads to a great significant performance decrease in absolute acc-mearsure (-1.39% and -2.13% on CoLA and STS-B), which demonstrates the effectiveness of the bidirectional interaction we set. We argue that the asymmetric attentional mechanisms between latent units and freezed representations mitigates the problems associated with

Factor s	Dim d'	CoLA	STS-B	QNLI	# Params
$s=1$	$d'=8$	72.26	91.97	94.02	1×
	$d'=16$	72.29	92.24	94.08	2×
$s=2$	$d'=16$	72.35	**92.45**	94.15	1×
	$d'=32$	**72.44**	92.36	**94.15**	2×
$s=4$	$d'=16$	72.16	91.89	93.74	0.5×
	$d'=32$	72.25	92.30	94.12	1×
$s=6$	$d'=24$	71.84	91.62	93.47	0.5×
	$d'=48$	72.17	92.18	93.85	1×

Table 5: The effect of different control chunking factors s as well as low dimensional mapping sizes d' on model performance. $1\times$ denotes the default model, which is essentially comparable to the parameters of the LoRA model with rank 8.

non-full-rank training, leading to promising performance. In addition, as show in Table 4, freezing the learning space of latent units can also be slightly detrimental to model performance (-0.11% and -0.17% in absolute acc-mearsure). This suggests that the units have also learned some information that is helpful for the model. We will analyze later if this information is relevant to the specific task.

However, **w/o projection** that not doing low-dimensional mapping does not significantly increase the performance of the model, and the results on CoLA do not change. This suggests that our mapping approach does not harm the model performance with reduced model parameters. We tried adding additional FFN modules and instead observed a decrease in model performance (-1.6% on STS-B and -0.51% on QNLI). This suggests that not more parameters lead to better performance. This is in line with the original intent of a series of parameter-efficient fine-tuning methods (He et al., 2022a; Zhang et al., 2023).

5.2 Analysis of Hyper-parameters

In this section, we focus on analyzing the impact of different hyperparameters on the model performance, including the number of latent units M, the factor controlling the chunking s, and the low-dimensional size of the projection d'.

Number of latent units Since the number of latent units is pre-fixed, we conduct a comparison experiment to find the suitable setting. We employ the experiment on three GLUE datasets with a range from 1 to 32. We can observe from Figure 4 that the model performance increases as the latent units grows. However, when the number of the latent units 8, the trend of performance improvement on all three datasets slows down, or even decreases slightly, suggesting that for these current sub-tasks, the representational power of 8 latent units is sufficient for modeling the relevant information needed for the task. Eventually, the latent units number M in the experiments is set to 8, which is deployed on all the other datasets.

Datasets	**AdaLoRA (s)**	**SoRA (s)**	**GloC (s)**
CoLA	160.2	57.2	110.4
SST-2	491.0	433.0	453.6
MRPC	27.3	24.8	26.2
STS-B	48.2	38.4	40.3
QNLI	1001.0	676.3	738.0
RTE	79.8	45.1	64.4
Avg.	301.3	212.5	238.8

Table 6: The average training time per epoch on six datasets. For each task, all experiments have the same batch size 32.

Chunking and hidden size To be explicit, excluding the number of parameters occupied by latent units, the number of model parameters stays the same when the rank of the LoRA model is $r = d'/s$. We observe in Table 5 that the results in the second row are generally higher than in the first row, suggesting that a larger d' leads to better model performance. The factor s controlling the chunking achieves the best results for a value of 2, suggesting that sparser information exchange leads to a decrease in model performance. For example, when $s = 6$, it brings a performance degradation of -0.27% on CoLA and -0.3% on QNLI, respectively.

5.3 Analysis of Training Cost

In addition to the additional introduction of the number of parameters, the state-of-art method (Zhang et al., 2023; Ding et al., 2023a) also takes into account the cost of training time. The results in Table 2 show that the proposed method achieves superior performance against LoRA and other methods which basically have the same number of parameters. From the results in the Table 6, the training time of the proposed method is comparable to other advanced methods. AdaLoRA Zhang et al. (2023) computing SVD introduces additional computational overhead, and SoRA's well-designed training schedule Ding et al. (2023a) leads to shorter time.

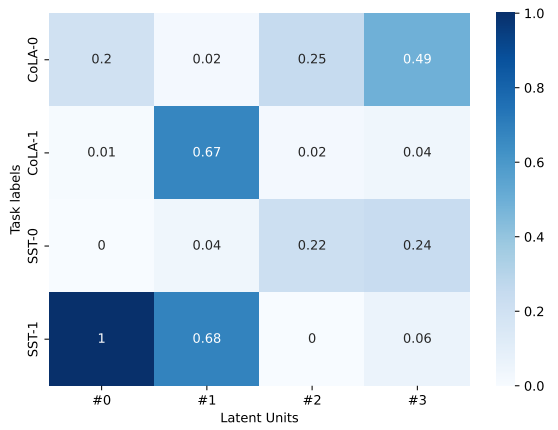

Figure 5: Co-occurrence statistics between the latent units and different task labels.

5.4 Analysis of Latent Units

Previous ablation experiments verified that the latent units learned features that contribute to the model's performance, and in this section we wish to verify that the latent units learned task-specific features, such as task topic content or task labeling information.

We count the co-occurrence of different latent units and different task labels on CoLA and SST-2 datasets. To eliminate the imbalance, we normalize the co-occurrence matrix first. As shown in Figure 5, different latent units have preferences for different task labels. For example, on CoLA task, the latent units #1 prefer to predict `CoLA-1` labels, while on SST-2 task, the latent units #2 and # 3 prefer to predict `SST-0` labels. These findings indicate that the latent units show strong statistical correlations with task labels, suggesting that the latent units learn task-specific relevant features.

6 Conclusion

In this paper, we present a novel global controller (GloC) approach for parameter-efficient fine-tuning. Based on a small set of latent units, we harness the large language model from a global perspective to seek optimal performance. We design asymmetric attention mechanisms and distillation compression modules during interaction to reduce training memory while mitigating the problem of non-full-rank training. Extensive experiments on a range of natural language understanding, generation, and reasnoning tasks show that our model reaches the state-of-the-art and significantly outperforms a range of robust baselines.

Limitations

We discuss here the limitations of the method in this paper. First, the proposed method has some hyperparameters, such as the number of latent units and the projection size. When the method needs to be migrated to a new task, parameter search is inevitably required. The second point is that the proposed methods have not been validated on ultra-large scale macromodels, such as Llama-70B, which we will validate in the subsequent work. The third point is that all PEFT methods inevitably have training memory bottlenecks due to the need for forward passes in the backbone model, and exploring new migration methods is a valuable direction.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62376245), the Key Research and Development Program of Zhejiang Province, China (No. 2024C03255), the Fundamental Research Funds for the Central Universities, and National Key Research and Development Project of China (No. 2018AAA0101900).

References

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *Proceedings of Text Analysis Conference*.

Srinadh Bhojanapalli, Chulhee Yun, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. 2020. Low-rank bottleneck in multi-head attention models. In *International conference on machine learning*, pages 864–873. PMLR.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and others. 2022. Palm:

Scaling language modeling with pathways. *ArXiv*, abs/2204.02311.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *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)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. 2023a. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4133–4145, Singapore. Association for Computational Linguistics.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023b. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *ICLR*.

Demi Guo, Alexander M Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *Proceedings of ACL*.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 7.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022a. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022b. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *Proceedings of ICML*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. 2021. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pages 3744–3753. PMLR.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of EMNLP*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of ACL*.

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

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John

Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Proceedings of NeurIPS*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Hugo Touvron and Louis Martin et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aur'elien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and Efficient Foundation Language Models. *ArXiv*, abs/2302.13971.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding.

Haowen Wang, Tao Sun, Cong Fan, and Jinjie Gu. 2023a. Customizable combination of parameter-efficient modules for multi-task learning.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023b. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Zonghan Yang and Yang Liu. 2022. On robust prefix-tuning for text classification. In *International Conference on Learning Representations*.

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

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. Glm-130b: An Open Bilingual Pre-trained Model. *ICLR 2023 poster*.

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

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open Pre-trained Transformer Language Models. *ArXiv*, abs/2205.01068.

Sanqiang Zhao, Raghav Gupta, Yang Song, and Denny Zhou. 2019. Extreme language model compression with optimal subwords and shared projections.

Method	# Params	BoolQ	PIQA	SIQA	HellaS	WinoG	OBQA	AVE.
LLaMA-Prompt	0.72%	65.1	75.6	71.4	43.6	72.4	58.3	64.4
LLaMA-Prefix	0.72%	64.3	76.8	73.9	42.1	72.1	60.6	65.0
LLaMA-Adapter	0.77%	67.9	76.4	78.8	69.8	78.9	75.2	74.5
LLaMA-LoRA	0.72%	68.9	80.7	77.4	78.1	76.8	74.8	76.1
LLaMA-AdaLora	0.69%	69.4	80.8	77.8	78.6	77.1	75.4	76.5
LLaMA-GloC	0.72%	72.1	83.6	78.7	81.4	79.2	77.1	78.7

Table 7: Baseline results with LLaMA on six common-sense reasoning datasets.

Dataset	Domain	# train	# test	Answer
BoolQ	CS	9.4K	3,270	Yes/No
PIQA	CS	16.1K	1,830	Option
SIQA	CS	33.4K	1,954	Option
HellaSwag	CS	39.9K	10,042	Option
WinoGrande	CS	63.2K	1,267	Option
OBQA	CS	5.0K	500	Option

Table 8: Details of datasets for commonsense reasoning.

A Appendix

A.1 Common-sense Reasoning Corpus

Additionally, for the reasoning task, we take six common-sense reasoning datasets, including:

(1) The BoolQ dataset (Clark et al., 2019) is a question-answering dataset for yes/no questions, consisting of 15,942 examples. These questions are naturally occurring and generated in unprompted and unconstrained settings.

(2) The PIQA dataset (Bisk et al., 2020) includes questions with two solutions that require physical commonsense to answer.

(3) The SIQA dataset (Sap et al., 2019) focuses on reasoning about people's actions and their social implications.

(4) The HellaSwag dataset consists of common-sense NLI questions that include a context and several possible endings to complete the context.

(5) The WinoGrande dataset (Sakaguchi et al., 2021) is formulated as a fill-in-the-blank task with binary options, where the goal is to choose the correct option for a given sentence requiring commonsense reasoning.

(6) The OBQA dataset includes questions that require multi-step reasoning, the use of additional common and commonsense knowledge, and rich text comprehension.

Table 8 shows the detailed dataset statistics. and possible answer options.

A.2 Other Baselines

Since prompt tuning has long training time and poor performance as mentioned in the Section 4 in Ding et al. (2023a), hence it was not included as the baseline in experimental section. For the sake of completeness in comparison and consistency with related work, we supplemented the results on the six common-sense reasoning datasets for prompt-tuning and prefix-tuning here.

We find that, with the same parameter numbers, prompt-tuning and prefix-tuning performed significantly worse than Adapter and LoRA on these six datasets, which further corroborates the claim made by Ding et al. (2023a).

A.3 Non-full-rank Training

Bhojanapalli et al. (2020) discussed the issue of low-rank bottleneck in multi-head attention, where the model's expressive power is constrained when the hidden size in each head is smaller than the context length. Differently, He et al. (2022b) discuss this issue in PEFT series approaches which was mentioned in their unified perspective of Section 3.1. When the rank r of the additional module is less than the hidden layer size d of the backbone model itself, it introduces a representational bottleneck.

Furthermore, in experiments, He et al. (2022b) found that this problem is more pronounced in the feed-forward network (FFN), thus allocating more parameters to FFN compared to the attention module to alleviate this issue. The theoretical explanation is that in multi-head attention, the hidden size d becomes d/N_h, so relatively, the representational bottleneck introduced by the rank r in attention diminishes. Our approach utilizes multi-head attention mechanism for information exchange, thus having the advantage of mitigating low-rank compared to LoRA-like bitwise summation methods. Compared to attention mechanisms such as prompt tuning and prefix tuning, the proposed method has an additional low-rank mapping space, which leads to better training convergence while mitigating expressive bottlenecks.