Mixture-of-Subspaces in Low-Rank Adaptation

Anonymous EMNLP submission

Abstract

In this paper, we introduce a *subspace*-inspired Low-Rank Adaptation (LoRA) method, which is computationally efficient, easy to implement, and readily applicable to large language, multimodal, and diffusion models. Initially, we equivalently decompose the weights of LoRA into two subspaces, and find that simply mixing them can enhance performance. To study such a phenomenon, we revisit it through a fine-grained subspace lens, showing that such modification is equivalent to employing a fixed *mixer* to fuse the subspaces. To be more flexible, we jointly learn the mixer with the original LoRA weights, and term the method as *Mixture-of-Subspaces LoRA (MoSLoRA)*. MoSLoRA consistently outperforms LoRA on tasks in different modalities, including commonsense reasoning, visual instruction tuning, and subject-driven text-to-image generation, demonstrating its effectiveness and robustness.

1 Introduction

Large Language Models (LLMs), such as GPT-4 (OpenAI, 2023), LLaMA 3 (AI@Meta, 2024), and InternLM2 (Cai et al., 2024), have demonstrated remarkable performance across diverse disciplines (Rozière et al., 2023; Thirunavukarasu et al., 2023). Such strong capability is often attributed to the increased scale of training data and model parameters. However, it also brings increasing challenges to adapting these LLMs for downstream tasks via fully fine-tuning all the parameters.

To tackle this issue, parameter-efficient fine-tuning (PEFT) has been developed (Hu et al., 2022; Lester et al., 2021; He et al., 2022) to minimize the number of optimized parameters while achieving comparable performance as much as possible. Among these methods, LoRA (Hu et al., 2022) has gained increasing popularity due to its simplicity and efficacy, which proposes to update the extra low-rank branch exclusively and merge it into the frozen original weight during inference. As shown in Figure 1, for the original weight matrix $\mathbf{W}_0 \in \mathbb{R}^{d_1 \times d_2}$, the additional low-rank branch consists of a down projection $\mathbf{A} \in \mathbb{R}^{d_1 \times r}$ and an up projection $\mathbf{B} \in \mathbb{R}^{r \times d_2}$, where $r \ll \min(d_1, d_2)$. Hence, the number of updated parameters is reduced from $d_1 \times d_2$ to $(d_1 + d_2)r$.

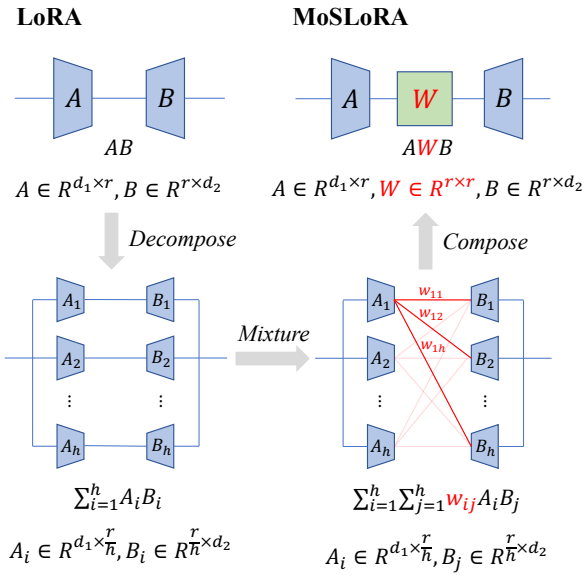


Figure 1: Comparison between vanilla LoRA and proposed MoSLoRA. In MoSLoRA, we employ learnable weights to mix more subspaces with negligible parameters (i.e. $(d_1 + d_2 + r)r$ vs $(d_1 + d_2)r$ and $d_1 + d_2 \gg r$ typically).

In this paper, we first define *subspaces* in LoRA as the parallel components with smaller rank values, similar to the subspace in multi-head attention (MHA) design (Vaswani et al., 2017). After that, we can decompose the vanilla LoRA into several subspaces via structural re-parameterization (Wu et al., 2023; Ding et al., 2021). Figure 2 indicates the process of decomposing into two subspaces. Interestingly, we find that simply mixing these two subspaces performs better in the commonsense reasoning tasks.

Motivated by the observation, we further revisit the two-subspaces-mixing strategy in a more fine-grained (rank=1) view and composed view. In short, such a strategy equals inserting a *mixer* matrix between $\mathbf{A}$ and $\mathbf{B}$, which is a fixed butterfly factor (Dao et al., 2019). Meanwhile, vanilla LoRA can be considered as a special case with a fixed identity matrix being the mixer. Therefore, we propose MoSLoRA, a simple yet effective method, which employs a learnable mixer to fuse more subspaces and more flexibly. As shown in Figure 1, we adapt the mixer $\mathbf{W}$ to fuse all the possible subspaces (i.e. A_iB_j). Compared to LoRA, MoSLoRA requires negligible extra parameters since $d_1 + d_2 \gg r$. Similarly to LoRA, MoSLoRA can also be merged into the original weights, and thus introduce no latency during inference.

We perform experiments on various downstream tasks, including commonsense reasoning tasks fine-tuning LLaMA 3 (AI@Meta, 2024), visual instruction tuning on LLaVA-1.5 (Liu et al., 2023a) series models, and subject-driven text-to-image generation on Stable Diffusion XL (SDXL) model (Podell et al., 2023). Experimental results indicate that the proposed MoSLoRA consistently outperforms LoRA and other baselines, demonstrating its effectiveness and robustness. Our contributions can be concluded as follows:

- We decompose LoRA into subspaces via structural re-parameterization, revealing a new pathway to investigate LoRA.
- We propose a simple yet effective MoSLoRA method, employing a learnable mixer to fuse more subspaces and more flexibly.
- We conduct extensive experiments on various downstream tasks, demonstrating the effectiveness and robustness of the proposed MoSLoRA.

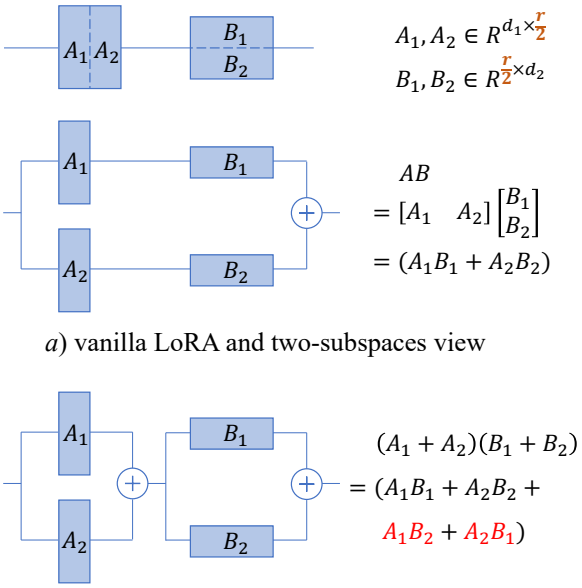
2 Preliminaries and Motivation

2.1 LoRA and Subspace View

Based on the hypothesis that the update in weights during model adaptation exhibits low intrinsic rank, LoRA (Hu et al., 2022) aims to model the weight update via two low-rank matrices. For a pre-trained weight matrix $\mathbf{W}_0 \in \mathbb{R}^{d_1 \times d_2}$ and arbitrary input x, they modify the forward pass as follows [1]:

$$x\mathbf{W}_0 + x\Delta\mathbf{W} = x\mathbf{W}_0 + x\mathbf{AB}, \quad (1)$$

[1] In this paper, we use the post-multiplication for simplicity.



a) vanilla LoRA and two-subspaces view

b) two-subspaces-mixing LoRA

Figure 2: Overview of decomposing vanilla LoRA into two subspaces and mixing them. Compared to vanilla LoRA, two-subspaces-mixing LoRA contains two extra entries.

where $\mathbf{A} \in \mathbb{R}^{d_1 \times r}$, $\mathbf{B} \in \mathbb{R}^{r \times d_2}$ and $r \ll \min(d_1, d_2)$. Typically, $\mathbf{A}$ is initialized as a Gaussian matrix and $\mathbf{B}$ as a zero matrix, so that $\Delta\mathbf{W}$ is zero at the beginning. During training, the original weight $\mathbf{W}_0$ is frozen, while $\mathbf{A}$ and $\mathbf{B}$ contain trainable parameters. After that, the $\mathbf{A}$ and $\mathbf{B}$ can be merged into $\mathbf{W}_0$ during inference, thus not introducing any latency.

In this paper, we decompose LoRA into subspaces via structural re-parameterization, where the subspaces are defined as parallel components with smaller rank values. Figure 2 part a shows the procedure for two subspaces. Specifically, we decompose the $\mathbf{A}$ into two parts (i.e. $\mathbf{A}_1$ and $\mathbf{A}_2$) by column, and $\mathbf{B}$ by row to get $\mathbf{B}_1$ and $\mathbf{B}_2$. Therefore, we can easily get that:

$$\begin{aligned} x\mathbf{AB} &= x \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \\ &= x(\mathbf{A}_1\mathbf{B}_1 + \mathbf{A}_2\mathbf{B}_2), \end{aligned} \quad (2)$$

where the $\mathbf{A}_1\mathbf{B}_1$ and $\mathbf{A}_2\mathbf{B}_2$ are the two subspaces. In the two-subspace view, vanilla LoRA equals the sum of two subspaces. Moreover, we can get a more fine-grained view if we split $\mathbf{A}$ and $\mathbf{B}$ for more parts, respectively.

2.2 Mixing Two Subspaces

As shown in Figure 2b, we can simply mix two subspaces by adding up the outputs of $\mathbf{A}_1$ and $\mathbf{A}_2$.

Method	ARC-e	OBQA	SIQA	ARC-c	WinoG.	PIQA	BoolQ	HellaS.	Avg.
LoRA (r=16)	87.7	82.8	79.3	75.7	84.8	86.7	72.3	93.5	82.8
+ TS-Mixing	88.3	83.0	80.3	78.1	84.8	87.5	73.8	94.3	**83.8**
LoRA (r=32)	83.5	82.6	80.3	70.3	82.6	85.7	71.3	91.4	81.0
+ TS-Mixing	87.9	84.2	79.9	75.1	84.8	86.9	72.1	93.3	**83.0**

Table 1: Comparison of vanilla LoRA and two-subspaces-mixing LoRA (denoted as TS-Mixing) on 8 benchmarks. Simply mixing these two subspaces leads to better performance.

Hence, the output of the whole module for input x would be:

$$\begin{aligned} &x(\mathbf{A}_1+\mathbf{A}_2)(\mathbf{B}_1+\mathbf{B}_2) \\ =&x(\mathbf{A}_1\mathbf{B}_1+\mathbf{A}_2\mathbf{B}_2+\mathbf{A}_1\mathbf{B}_2+\mathbf{A}_2\mathbf{B}_1). \end{aligned} \tag{3}$$

Compared to Equation 2, Equation 3 contains two extra entries and can model more information intuitively.

To compare these two strategies, we conduct experiments on the commonsense reasoning tasks following Hu et al. (2023). We first fine-tune LLaMA-3 8B model (AI@Meta, 2024) on 170k training samples (Hu et al., 2023), and then report the performance on 8 benchmarks, including ARC-c/e (Clark et al., 2018), OBQA (Mihaylov et al., 2018), SIQA (Sap et al., 2019), WinoG. (WinoGrande) (Sakaguchi et al., 2020), PIQA (Bisk et al., 2020), BoolQ (Clark et al., 2019), and HellaS. (HellaSwag) (Zellers et al., 2019). Please refer to Appendix A.1 for details of these benchmarks. All hyperparameters are the same and listed in Appendix B.1.

Table 1 shows the results on 8 benchmarks for these two methods. Mixing two subspaces would lead to better performance under different settings (r=8/16), such as 93.3 compared to 91.4 of LoRA on the HellaSwag benchmark, showing the effectiveness and robustness of two-subspaces-mixing LoRA than vanilla LoRA.

3 Methodology

3.1 More Fine-grained Subspace

Motivated by the observation that mixing two subspaces would lead to better performance, we revisit the two-subspaces-mixing LoRA in view of more fine-grained subspace (i.e. rank=1). Specifically, we decompose the $\mathbf{A}\in\mathbb{R}^{d_1\times r}$ and $\mathbf{B}\in\mathbb{R}^{r\times d_2}$ into r subspaces (rank=1), which can be formulated as:

$$\begin{aligned} \mathbf{A}&=\begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_r\end{bmatrix} \\ \mathbf{B}^T&=\begin{bmatrix}\mathbf{B}_1^T & \mathbf{B}_2^T & \cdots & \mathbf{B}_r^T\end{bmatrix}, \end{aligned} \tag{4}$$

Method	#N of subspaces (rank=1)	Trainable
LoRA	r	✗
TS-Mixing	$2r$	✗
MoSLoRA	r^2	✓

Table 2: Comparison of LoRA, two-subspaces-mixing LoRA (denoted as TS-Mixing), and proposed MoSLoRA. #N denotes the number of mixed subspaces.

where $\mathbf{A}_i\in\mathbb{R}^{d_1\times 1}$ and $\mathbf{B}_i\in\mathbb{R}^{1\times d_2}$ for $1\le i\le r$. As shown in Figure 3, we can thus view vanilla LoRA as:

$$x\mathbf{AB}=x\sum_{i=1}^{r}\mathbf{A}_i\mathbf{B}_i=x\mathbf{A}\mathbf{I}_{r\times r}\mathbf{B}. \tag{5}$$

The $\mathbf{I}_{r\times r}\in\mathbb{R}^{r\times r}$ denotes the identity matrix. Meanwhile, the two-subspaces-mixing LoRA equals to:

$$\begin{aligned} &x\sum_{i=1}^{r/2}(\mathbf{A}_i+\mathbf{A}_{i+r/2})(\mathbf{B}_i+\mathbf{B}_{i+r/2}) \\ =&x\mathbf{A}\begin{bmatrix}\mathbf{I}_{r/2\times r/2} & \mathbf{I}_{r/2\times r/2}\\ \mathbf{I}_{r/2\times r/2} & \mathbf{I}_{r/2\times r/2}\end{bmatrix}\mathbf{B}. \end{aligned} \tag{6}$$

Interestingly, we can find that Equation 5 and Equation 6 share the same paradigm:

$$\mathbf{AWB}, \tag{7}$$

where $\mathbf{W}\in\mathbb{R}^{r\times r}$ and we define $\mathbf{W}$ as the weight of **mixer** to fuse the subspaces. For vanilla LoRA, the mixer is the fixed identity matrix fusing r subspaces. For the two-subspaces-mixing LoRA, the mixer is a fixed butterfly factor fusing $2r$ subspaces, which is more than LoRA. Therefore, we propose MoSLoRA, adapting a trainable mixer to fuse all the possible subspaces. As shown in Table 2, MoSLoRA mixes the information of r^2 subspaces (rank=1) employing trainable weights, modeling the information of more subspaces and more flexible than LoRA.

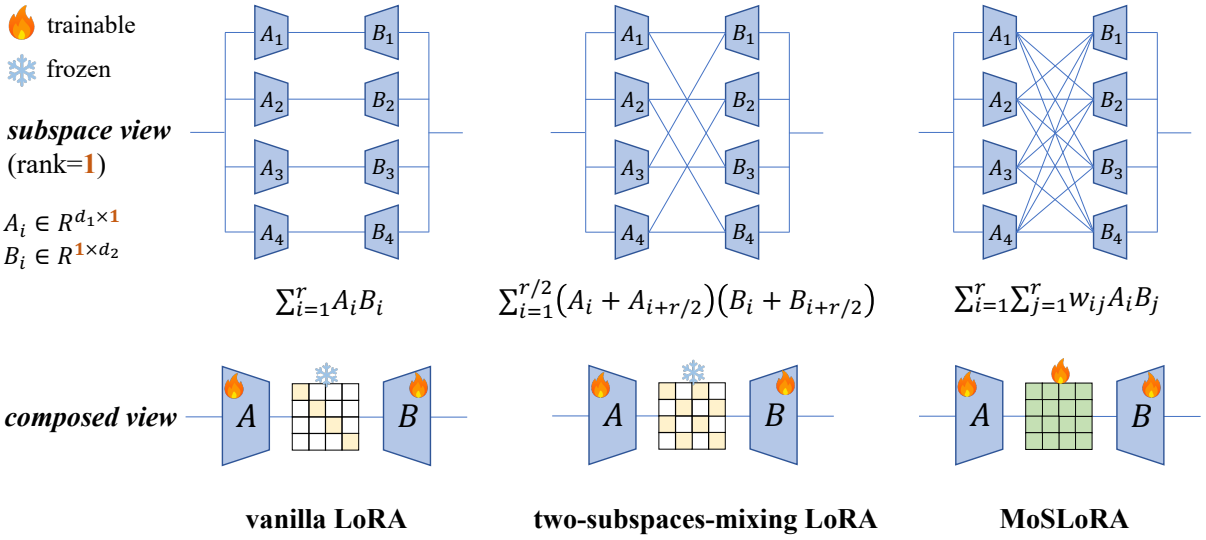


Figure 3: The subspace view (rank=1) and composed view for vanilla LoRA, two-subspaces-mixing LoRA, and proposed MoSLORA. In MoSLoRA, we employ a learnable mixer to fuse more information and more flexibly.

Initialization Strategy	Average Score
Zero Matrix	*not converge*
Identity Matrix	82.6
Normal Distribution	80.7
Orthogonal Matrix	84.4
Kaiming Uniform Distribution	85.6

Table 3: Comparison of various initialization strategies for the trainable mixer in MoSLoRA. We report the average score on the commonsense reasoning tasks.

3.2 Initialization Strategies for Mixer

In the proposed MoSLoRA, we employ a trainable mixer to fuse all possible subspaces. However, the system of MoSLoRA is linear, and thus a bad initialization can still hamper the learning (He et al., 2015). In MoSLoRA, we follow the setting in LoRA and initialize $\mathbf{A}$ using a *Kaiming uniform* distribution [2] and $\mathbf{B}$ as a *zero* matrix. For the mixer weight $\mathbf{W}$, we compare various initialization strategies, including zero matrix, identity matrix, normal distribution, orthogonal matrix (Saxe et al., 2014), and Kaiming uniform distribution (He et al., 2015). Hyperparameters for finetuning can be found at Appendix B.1.

Table 3 reports the results of the commonsense reasoning tasks. If we initialize the mixer as the zero matrix, then the model would not converge since all of the $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{W}$ get zero gradients (cf. Appendix C for proof). When initializing the mixer as an identity matrix and updating it during training, the performance is similar to the vanilla LoRA with a fixed identity (82.6 vs. 82.8). Moreover, Kaiming uniform distribution and orthogonal matrix get strong performance, and thus we adapt them for the initialization of the mixer in MoSLoRA.

[2] In the code of LoRA, they use Kaiming uniform initialization rather than Gaussian distribution claimed in the paper.

3.3 Relation with Mixture-of-Experts

Mixture-of-Experts (MoE) methods aim to partition a set of parameters into experts and route input samples to specific experts during training and inference (Fedus et al., 2022a). Typically, they employ a router to generate scores for each expert based on the input, and then select top-k experts (Fedus et al., 2022b; Lepikhin et al., 2021; DeepSeek-AI, 2024). In this paper, we propose MoSLoRA to mix the subspaces in LoRA, where the w_{ij} in the mixer can be considered as the weight to compose subspace $\mathbf{A}_i\mathbf{B}_j$. However, the differences between MoSLoRA and MoE methods are as follows:

- In MoSLoRA, the weights to mix subspaces are input agnostic, while weights from gates in MoE methods are input specific.
- In MoSLoRA, we adapt all the subspaces simultaneously, while MoE methods select top-k from all the experts.

Method	Param	Time	Mem	ARC-e	OBQA	SIQA	ARC-c	WinoG.	PIQA	BoolQ	HellaS.	Avg.
LoRA	28.3M	8.0h	29G	87.7	82.8	79.3	75.7	84.8	86.7	72.3	93.5	82.8
LoKr	0.9M	26.3h	66G	89.2	81.8	78.7	76.7	82.1	81.6	65.1	92.0	80.9
LoHa	28.3M	25.5h	68G	91.2	85.8	81.1	80.5	83.3	89.7	75.0	95.0	85.2
FLoRA	28.4M	8.2h	31G	90.2	84.2	79.9	79.3	85.1	86.7	74.8	93.9	84.2
AdaLoRA	28.3M	12.5h	58G	90.4	85.0	76.7	79.1	83.3	86.4	75.1	75.4	81.4
DoRA	29.1M	14.5h	33G	90.1	87.2	80.3	79.1	84.7	88.8	74.5	95.5	85.0
DoRA*	57.4M	14.8h	33G	90.5	85.8	79.9	80.4	85.6	89.3	74.6	95.5	85.2
MoSLoRA	28.4M	8.2h	31G	90.5	86.8	81.0	81.5	85.8	89.7	74.6	95.0	**85.6**

Table 4: Accuracy comparison of various methods fine-tuning LLaMA-3 8B on the commonsense reasoning tasks. **Param** denotes the number of trained parameters, **Time** for the training time on A100 GPU, and **Mem** for the GPU Memory usage. * denotes a larger rank in DoRA. We can find that the proposed MoSLoRA outperforms all the baselines with a slightly extra training cost than LoRA.

4 Experiments and Analysis

4.1 Commonsense Reasoning

We fine-tune LLaMA-3 8B instruction version model (AI@Meta, 2024) for the commonsense reasoning question answering tasks. We first train the model using 170k training samples (Hu et al., 2023), and then test the fine-tuned model on 8 commonsense reasoning question answering benchmarks (refer to Appendix A.1 for details). The 170k training set is the mixture of the training sets of these benchmarks. Besides LoRA (Hu et al., 2022), we also compare MoSLoRA with various baselines, including: 1) LoKr (Yeh et al., 2023) which employs Kronecker products for matrix decomposition of $\mathbf{AB}$; 2) LoHa (Yeh et al., 2023) which decomposes the vanilla LoRA into the Hadamard product of two LoRA branches; 3) FLoRA (Si et al., 2024) which introduces an extra core based on Tucker decomposition to maintain the consistent topological structure with the original space [3]; 4) AdaLoRA (Zhang et al., 2023) which parameterizes the incremental updates of the pre-trained weight matrices in the form of singular value decomposition; and 5) DoRA (Liu et al., 2024) which decomposes the pretrained weight into its magnitude and directional components and fine-tunes both of them.

All the experiments are conducted using 1 Nvidia 80G A100 GPU. The hyperparameters are listed in Appendix B.1. Based on the analysis in Table 3, we initialize the mixer following the Kaiming uniform distribution. Besides the accuracy, we also report the number of trained parameters and training overhead including time and peak GPU memory.

[3]Please refer to Appendix D for the discussion of differences.

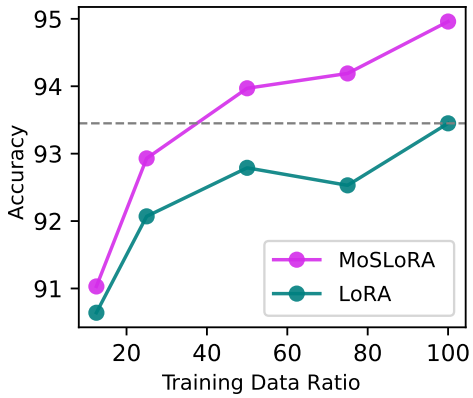


Figure 4: Comparison of MoSLoRA and LoRA on the HellaSwag benchmark with fewer training samples.

Table 4 shows the results on 8 benchmarks. Some findings can be summarized as follows:

- MoSLoRA outperforms all the baselines, demonstrating the effectiveness of mixing the subspaces. Specifically, MoSLoRA gets an average of 85.6, which is 2.8 higher than the 82.8 of LoRA. Moreover, MoSLoRA outperforms DoRA with a higher rank.
- Compared to LoRA, MoSLoRA requires *negligible* extra parameters (less than 0.1M) and computing cost (less than 0.2h). Meanwhile, MoSLoRA can save 44% training time than DoRA and 68% than LoHa.
- Though LoKr reduces the training parameters via Kronecher products, it requires more than 3x training time and 2x GPU memory than MoSLoRA. Also, LoKr gets an average score of 80.9, which is 4.7 lower than MoSLoRA.

Fewer training samples To compare the performance under fewer sample settings, we randomly

Model	Method	Init.	MMBench EN	MMBench CN	SEED-Bench	AI2D	SciQA image	Text VQA	Math Vista	MM-Vet	MME	Avg.
LLaMA-3 +ViT	LoRA	-	72.0	67.8	68.8	61.4	74.8	47.1	27.7	33.1	58.4	56.8
	MoSLoRA	*Orth*	73.0	68.2	69.0	61.2	75.7	47.2	27.6	33.4	60.6	**57.3**
		Kai	72.5	67.5	68.9	60.6	76.0	47.1	27.5	33.8	60.5	57.1
InternLM2 +ViT	QLoRA	-	70.8	68.9	70.4	62.2	72.5	49.8	30.2	33.9	61.6	57.8
	QMoSLoRA	*Orth*	73.5	71.2	71.1	64.8	71.8	49.8	30.2	35.0	62.0	58.8
		Kai	73.8	72.6	70.3	66.1	72.2	50.2	30.6	35.2	64.1	**59.5**

Table 5: Results on 9 benchmarks for vanilla LoRA and proposed MoSLoRA. In MoSLoRA, we try both orthogonal (denoted as *Orth*) and Kaiming uniform initialization (denoted as *Kai*). For InternLM2, we employ the 4-bit QLoRA on LoRA and MoSLoRA. MoSLoRA consistently outperforms LoRA on various backbones for both initialization strategies.

select 12.5%/25%/50%/75% training samples from the original 170k training set and repeat the experiments. As shown in Figure 4, more training samples would lead to better performance and MoSLoRA outperforms LoRA under all the settings. Particularly, MoSLoRA trained via 50% samples gets a score of 83.6, which is 1.8 higher than LoRA using 100% samples. Moreover, the performance gap between MoSLoRA and LoRA becomes larger as the training samples increase, showing the superiority of MoSLoRA to modeling more complex information due to the mixture of subspaces.

4.2 Visual Instruction Tuning

To evaluate performance on multimodal tasks, we fine-tune the LLaVA-1.5 (Liu et al., 2023a) series models for visual instruction tuning, and then test the model for various visual QA benchmarks.

There are two stages in training LLaVA: 1) pretrain a two-layer MLP to project visual features to language space, and 2) optimize LLM and visual encoder (optional) for visual instruction tuning. In this paper, we employ the pretrained projector provided in XTuner (Contributors, 2023b), and then conduct visual instruction tuning on the LLM backbone and visual encoder, simultaneously. For the LLM backbones, we adapt the LLaMA3 8B (AI@Meta, 2024) and InternLM2 7B (Cai et al., 2024) using the off-the-shelf projecters [4]. For the visual encoder, we employ the ViT [5] (Dosovitskiy et al., 2021) large version. Due to limited resources, we fintune both the LLM backbone and visual encoder via LoRA/MoSLoRA on the 665K instruction-following data (Liu et al., 2023a), rather than optimize all the parameters in LLMs. For InternLM2, we employ the 4-bit QLoRA (Dettmers et al., 2023) and corresponding QMoSLoRA (QLoRA+MoSLoRA). Based on the results in Table 3, we initialize the mixer as the orthogonal matrix and Kaiming uniform distribution, separately. For specific hyperparameters, please refer to the Appendix B.2. It takes around 20 hours to fine-tune using 4 Nvidia A100 80G GPUs.

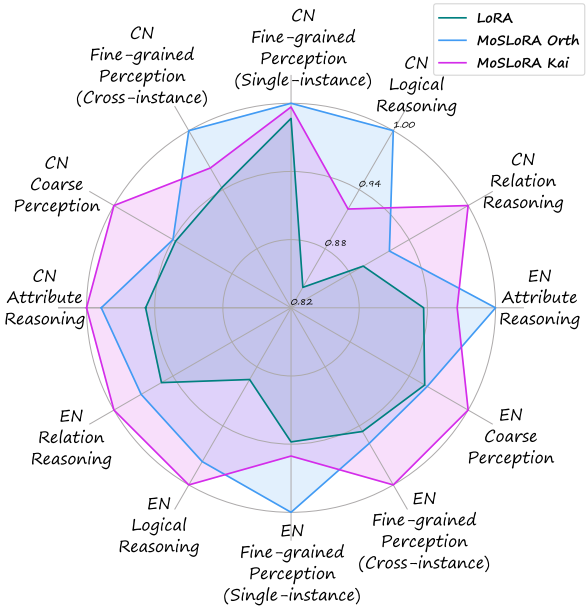


Figure 5: Normalized performance on 6 ability dimensions in MMBench EN/CN for QLoRA and QMoSLoRA when fintuning InternLM2. MoSLoRA significantly improves the reasoning ability over LoRA.

After visual instruction tuning, we evaluate the trained model on 9 popular benchmarks, including MMBench EN/CN (Liu et al., 2023b), SEED Bench (Li et al., 2023), AI2D (Kembhavi et al., 2016), SciQA (Lu et al., 2022), TextVQA (Singh et al., 2019), MathVista testmini (Lu et al., 2023),

[4] pretrained projecters
[5] openai/clip-vit-large-patch14-336

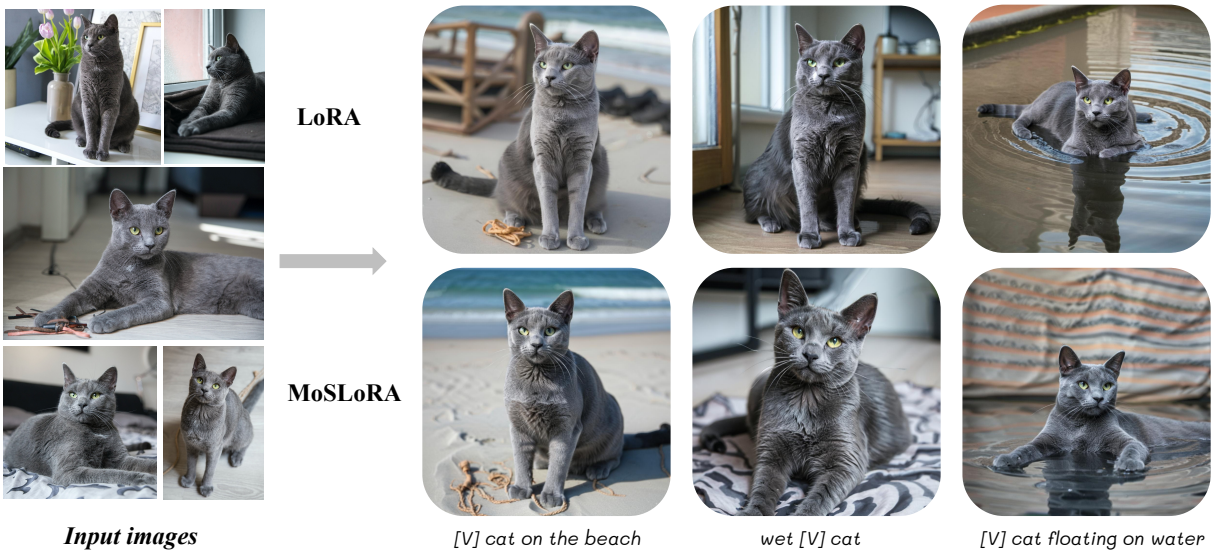


Figure 6: Comparison of generated images from LoRA and MoSLoRA on the subject-driven generation task. MoSLoRA is more consistent with the subject in the input images (e.g. the color of the hairs around the neck) and conforms to the given prompts (e.g. the wet hair and floating gesture) better.

MM-Vet (Yu et al., 2023), and MME (Fu et al., 2023). All the evaluations are done using the VLMEvalKit (Contributors, 2023a). Please refer to Appendix A.2 for the details of the dataset and the reported metrics. Specifically, we scale the MME scores to 100 to calculate the average score.

Table 5 shows the results on 9 benchmarks. For both orthogonal and Kaiming initialization, MoSLoRA consistently outperforms LoRA on various benchmarks. Specifically, MoSLoRA gets an average score of 59.5 on InternnLM2+ViT, which is 1.7 higher than LoRA. Moreover, MoSLoRA also outperforms LoRA when combined with the 4-bit QLoRA. It effectively showcases the compatibility of MoSLoRA with QLoRA. Therefore, MoSLoRA can be applied in low-resource finetuning scenarios combined with the quantization methods. In summary, the proposed MoSLoRA consistently outperforms LoRA in various settings, demonstrating its effectiveness and robustness.

More finegrained ability Moreover, we also visualize the normalized scores on 6 ability dimensions in the MMbench EN/CN test set. As shown in Figure 5, we can observe that MoSLoRA performs better than LoRA on all abilities for both English and Chinese scenarios, especially the reasoning ability. Reasoning tasks are typically considered to be more complex and difficult. Compared to LoRA, MoSLoRA mixes more subspaces and is thus better at more difficult tasks such as logical reasoning.

4.3 Subject-driven Generation

We further perform the experiments fine-tuning the text-to-image diffusion models for the subject-driven generation task (Ruiz et al., 2023). The goal is to generate the images following the given prompts of one specific subject, which is defined in a few given images. We first fine-tune a text-to-image model with the input images paired with a text prompt containing a unique identifier (e.g., A photo of a [V] cat). After that, we can employ other prompts containing the unique identifier to generate the corresponding images.

Figure 6 shows one case of a dog from the DreamBooth dataset (Ruiz et al., 2023). We fine-tune the SDXL[6] model (Podell et al., 2023) via LoRA and MoSLoRA. In MoSLoRA, the mixer is initialized as an orthogonal matrix. During finetuning, the learning rate is 1e-4, and the batch size is 4. We train the model for 500 steps, which costs around 16 minutes using 1 80G A100 GPU. During generation, we infer 50 steps for the given prompts. Compared to vanilla LoRA, we can find that our proposed MoSLoRA captures more details of the subject and better conforms to the given prompt. Specifically, MoSLoRA learns more details about the given cat, including the color of the hairs around the neck and the shape of the paw. Meanwhile, the images from MoSLoRA are more consistent with the given prompts, such as the wet (thus clumped) hair and the floating gesture (spread hands).

[6] stable-diffusion-xl-base-1.0

Metric	Win	Tie	Loss	Δ
Sub-simi	23.1%	60.4%	16.5%	+6.6%
Pro-cons	45.1%	44.2%	10.7%	+34.3%

Table 6: Human evaluation results on the generated images comparing MoSLoRA against LoRA. *Sub-simi* denotes for the subject similarity and *Pro-cons* for prompt consistency.

Human evaluation We also perform human evaluation on the generated images. First, we choose four subjects (i.e. cat, dog, grey sloth plushie, and can) from the DreamBooth dataset (Ruiz et al., 2023) and fine-tune the SDXL model, respectively. Then, we randomly select 8 prompts to generate the corresponding images. After that, 15 human experts are asked to independently score win/tie/loss for the paired images from LoRA and MoSLoRA. During evaluation, we shuffle these pairs and keep that these experts do not know the source model of each image. We employ two metrics, including 1) *subject similarity* defined as the similarity between subjects from generated images and given images, and 2) *prompt consistency* defined as the consistency among prompts and generated images. Table 6 reports the average score for all the images. We can find that MoSLoRA outperforms LoRA on both metrics. In particular, MoSLoRA gets an average winning ratio of 45.1% on prompt consistency, which is 34.3% than LoRA. Please refer to Appendix E for the detailed prompts and corresponding generated images from LoRA and MoSLoRA.

5 Related Work

5.1 Parameter-Efficient Fine-tuning

Parameter-efficient fine-tuning (PEFT), aiming to update a small proportion of parameters to adapt Large Language Models (LLMs), has become increasingly important. The mainstreaming PEFT methods can be categorized into: 1) adapter based methods (Houlsby et al., 2019; Lei et al., 2023), which inserts modules between transformer layers; 2) prefix tuning methods (Li and Liang, 2021; Liu et al., 2021), which prepends tunable prefix vectors into the hidden states; 3) selective methods (Zaken et al., 2022), which select part of the parameters to update; and 4) low-rank adapting (LoRA) series (Hu et al., 2022; Yeh et al., 2023), which injects trainable low-rank branches to approximate the weight updates. In LoRA, low-rank branches can be merged into the original weights during inference, thus bringing no latency. We refer the reader to Han et al. (2024) for a more comprehensive survey. In this paper, we focus on LoRA methods.

5.2 LoRA and its Variants

The core of LoRA is to update the mergeable and low-rank branches to model the weight updates. Hu et al. (2022) initialize the branch as a product of two low-rank matrices. The following variants can be categorized into: 1) *introducing training skills*, such as setting different learning rates (Hayou et al., 2024) and adding random noise (Lin et al., 2024); 2) *searching ranks*, such as DyLoRA (Valipour et al., 2023) and AdaLoRA (Zhang et al., 2023); and 3) *new designs* for the branch, such as LoKr (Yeh et al., 2023), LoHa (Yeh et al., 2023), VeRA (Kopiczko et al., 2023) and (Liu et al., 2024). LoKr and LoHa employ Kronecker and Hadamard products to replace the vanilla matrix product, respectively. DoRA decomposes the pretrained weight into its magnitude and directional components and fine-tunes them separately.

In this paper, we decompose LoRA into subspaces via structural re-parameterization and design a learnable mixer to fuse information from more subspaces and more flexibly.

6 Conclusion

This work proposes a novel MoSLoRA method for parameter-efficient fine-tuning. We first decompose the LoRA into subspaces and find that simply mixing the half-rank subspaces would lead to better performance. After that, we revisit vanilla LoRA and two-subspaces-mixing strategy in a more fine-grained view (i.e. rank=1), thus unifying both methods as employing an extra fixed mixer. Therefore, we propose MoSLoRA, which employs a learnable mixer to fuse more information and more flexibly. The mixer requires negligible extra parameters and computing costs. Experimental results on commonsense reasoning tasks, visual instruction tuning tasks, and subject-driven generation tasks demonstrate the effectiveness and robustness of the proposed MoSLoRA. For future work, we would consider applying MoSLoRA for more tasks. Finding a task-specific way to initialize the mixer for faster convergence would be another interesting topic.

Limitations

In this paper, we conduct experiments on commonsense reasoning tasks, visual instruction tuning tasks, and subject-driven generation tasks. LoRA can be applied in more scenarios, such as mixing styles in image generation tasks when fine-tuning stable diffusion models. We leave these tasks for future work.

Ethics Statement

This project aims to improve the LoRA methods and can be employed for subject-driven text-to-image generation tasks, where the users can fine-tune the stable diffusion models to generate images of a specific subject defined by the input images. In some cases, such malicious parties might use the generated images to mislead viewers. This is a common issue in generative model approaches or content manipulation techniques.

References

AI@Meta. 2024. Llama 3 model card.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, and et al. 2024. Internlm2 technical report. *CoRR*, abs/2403.17297.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

OpenCompass Contributors. 2023a. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

XTuner Contributors. 2023b. Xtuner: A toolkit for efficiently fine-tuning llm. https://github.com/InternLM/xtuner.

Tri Dao, Albert Gu, Matthew Eichhorn, Atri Rudra, and Christopher Ré. 2019. Learning fast algorithms for linear transforms using butterfly factorizations. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1517–1527. PMLR.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Xiaohan Ding, Tianxiang Hao, Jianchao Tan, Ji Liu, Jungong Han, Yuchen Guo, and Guiguang Ding. 2021. Resrep: Lossless CNN pruning via decoupling remembering and forgetting. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 4490–4500. IEEE.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

William Fedus, Jeff Dean, and Barret Zoph. 2022a. A review of sparse expert models in deep learning. *CoRR*, abs/2209.01667.

William Fedus, Barret Zoph, and Noam Shazeer. 2022b. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. 2023. MME: A comprehensive evaluation benchmark for multimodal large language models. *CoRR*, abs/2306.13394.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *CoRR*, abs/2403.14608.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. 2024. Lora+: Efficient low rank adaptation of large models. *CoRR*, abs/2402.12354.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1026–1034. IEEE Computer Society.

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 the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5254–5276. Association for Computational Linguistics.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Min Joon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pages 235–251. Springer.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki Markus Asano. 2023. Vera: Vector-based random matrix adaptation. *CoRR*, abs/2310.11454.

Tao Lei, Junwen Bai, Siddhartha Brahma, Joshua Ainslie, Kenton Lee, Yanqi Zhou, Nan Du, Vincent Y. Zhao, Yuexin Wu, Bo Li, Yu Zhang, and Ming-Wei Chang. 2023. Conditional adapters: Parameter-efficient transfer learning with fast inference. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023. Seed-bench: Benchmarking multimodal llms with generative comprehension. *CoRR*, abs/2307.16125.

Xiang Lisa Li and Percy Liang. 2021. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Yang Lin, Xinyu Ma, Xu Chu, Yujie Jin, Zhibang Yang, Yasha Wang, and Hong Mei. 2024. Lora dropout as a sparsity regularizer for overfitting control. *CoRR*, abs/2404.09610.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning. *CoRR*, abs/2310.03744.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. Dora: Weight-decomposed low-rank adaptation. *CoRR*, abs/2402.09353.

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

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2023b. Mmbench: Is your multi-modal model an all-around player? *CoRR*, abs/2307.06281.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating math reasoning in visual contexts with gpt-4v, bard, and other large multimodal models. *CoRR*, abs/2310.02255.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2381–2391. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. SDXL: improving latent diffusion models for high-resolution image synthesis. *CoRR*, abs/2307.01952.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton-Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code. *CoRR*, abs/2308.12950.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 22500–22510. IEEE.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *CoRR*, abs/1904.09728.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. 2014. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Chongjie Si, Xuehui Wang, Xue Yang, Zhengqin Xu, Qingyun Li, Jifeng Dai, Yu Qiao, Xiaokang Yang, and Wei Shen. 2024. Flora: Low-rank core space for n-dimension. *arXiv preprint arXiv:2405.14739*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards VQA models that can read. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8317–8326. Computer Vision Foundation / IEEE.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature medicine*, 29(8):1930–1940.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2023. Dylora: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 3266–3279. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Taiqiang Wu, Cheng Hou, Zhe Zhao, Shanshan Lao, Jiayi Li, Ngai Wong, and Yujiu Yang. 2023. Weight-inherited distillation for task-agnostic BERT compression. *CoRR*, abs/2305.09098.

Shin-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard B. W. Yang, Giyeong Oh, and Yanmin Gong. 2023. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. *CoRR*, abs/2309.14859.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *CoRR*, abs/2308.02490.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1–9. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

A Details of Benchmarks

A.1 Commonsene Reasoning

The details of the benchmarks are as follows:

- ARC-c/e (Clark et al., 2018): the Challenge Set and Easy Set of ARC dataset of genuine grade-school level, containing 2376/1172 multiple-choice science questions in the test set, respectively.
- OBQA (Mihaylov et al., 2018): questions requiring multi-step reasoning, use of additional commonsense knowledge, and rich text comprehension. There are 500 questions in the test set.
- SIQA (Sap et al., 2019): reasoning questions about people's actions and their social implications. There are 1954 questions in the test set.
- WinoG. (WinoGrande) (Sakaguchi et al., 2020): fill-in-a-blank task with binary options to choose the right option for a given sentence which requires commonsense reasoning. There are 1267 questions in the test set.
- PIQA (Bisk et al., 2020): questions with two solutions requiring physical commonsense. There are 1830 questions in the test set.
- BoolQ (Clark et al., 2019): yes/no questions which are naturally occurring and generated in unprompted and unconstrained settings. There are 3270 questions in the test set.
- HellaS. (HellaSwag) (Zellers et al., 2019): commonsense NLI questions including a context and several endings which complete the context. There are 10042 questions in the test set.

For all the benchmarks, we report the accuracy following Hu et al. (2023).

A.2 Visual Instruction Tuning

The details of benchmarks and reported metrics are as follows:

- MMBench EN/CN (Liu et al., 2023b): the English and Chinese version of MMBench. MMBench contains over 3000 multiple-choice questions covering 20 different ability dimensions. Each ability dimension encompasses over 125 questions. We report the accuracy of the *test* set [7].
- SEED Bench (Li et al., 2023): 19K multiple choice questions with accurate human annotations, which spans 12 evaluation dimensions including the comprehension of both the image and video modality. In this paper, we use the image modality only and report the accuracy.
- AI2D (Kembhavi et al., 2016): AI2 Diagrams (AI2D) of over 5000 grade school science diagrams and more than 15000 corresponding multiple choice questions. We report the accuracy of the test set.
- SciQA (ScienceQA) (Lu et al., 2022): 21k multimodal multiple choice questions with diverse science topics and annotations of their answers with corresponding lectures and explanations. We report the accuracy of the test set.
- TextVQA (Singh et al., 2019): 45,336 questions on 28,408 images that require reasoning about text to answer. We report the accuracy of the validation set.
- MathVista testmini (Lu et al., 2023): a benchmark designed to combine challenges from diverse mathematical and visual tasks. It consists of 6,141 examples, derived from 28 existing multimodal datasets involving mathematics and 3 newly created datasets. We report the accuracy scores on the testmini subset of 1,000 examples using GPT-4-turbo.
- MM-Vet (Yu et al., 2023): 200 images and 218 questions (samples), including 187 images from various online sources with 205 questions, 10 images from VCR with 10 paired questions, and 3 paired questions and images for medical expert knowledge. We report the average scores from the GPT-4-turbo.
- MME (Fu et al., 2023): 14 subtasks aiming to measure both perception and cognition abilities and the answer is yes or no. For the metrics, original scores include accuracy and accuracy+ for each task, and the total score is 2800. In this paper, we *scale* the scores to 100 for average.

[7] Online submission for results

Hyperparameter	LoRA	LoKr	LoHa	FLoRA	AdaLoRA	MoSLoRA	DoRA	DoRA*
Rank r	16							32
α	32							64
Dropout	0.05							
Batch size	16							
Epochs	3							
Learning rate	3e-5						1e-5	
Target module	q, k, v, up, down							

Table 7: The hyperparameters for various methods on the commonsense reasoning tasks.

Hyperparameter	LLaMA-3+ViT	InternLM2+ViT
Batch size	8	16
Accumulative	2	1
Learning rate	2e-5	
Epoch	1	
Rank r	64/64	512/64
α	128/16	256/16
Target module	q, k, v, o, up, down, gate	

Table 8: The hyperparameters for various methods for visual instruction tuning. For rank and alpha, we report in the format of LLM/Visual Encoder.

B Experimental Setup

B.1 Commonsene Reasoning

Table 7 shows the detailed hyper-parameters for commonsense reasoning tasking when fine-tuning the LLaMA3-8B instruction version. For AdaLoRA, we set both the initial rank and target rank to be 16.

B.2 Visual Instruction Tuning

Table 8 reports the detailed hyper-parameters for visual instruction tuning when fine-tuning the LLaMA3-8B+ViT and InternLM2+ViT. Moreover, we employ the 4-bit QLoRA when finetuning the InternLM2, where the quantization type is NF4 with double quantization skills.

C Initialize Mixer as Zero Matrix

In MoSLoRA, we model the forward process as:

$$\begin{aligned} y &= x\mathbf{W}_{merge} \\ \mathbf{W}_{merge} &= \mathbf{W}_0 + \mathbf{AWB}, \end{aligned} \tag{8}$$

where the $\mathbf{W}_0$ is frozen during training. Then we have:

$$\begin{aligned} \frac{\partial y}{\partial \mathbf{A}} &= \frac{\partial y}{\partial \mathbf{W}_{merge}} \mathbf{B}^T \mathbf{W}^T \\ \frac{\partial y}{\partial \mathbf{W}} &= \mathbf{A}^T \frac{\partial y}{\partial \mathbf{W}_{merge}} \mathbf{B}^T \\ \frac{\partial y}{\partial \mathbf{B}} &= \mathbf{W}^T \mathbf{A}^T \frac{\partial y}{\partial \mathbf{W}_{merge}} \end{aligned} \tag{9}$$

If we initialize $\mathbf{W}$ and $\mathbf{B}$ as zero matrices simultaneously, all the gradients in Equation 9 would be zero, and neither would be updated.

D Differences with FLoRA

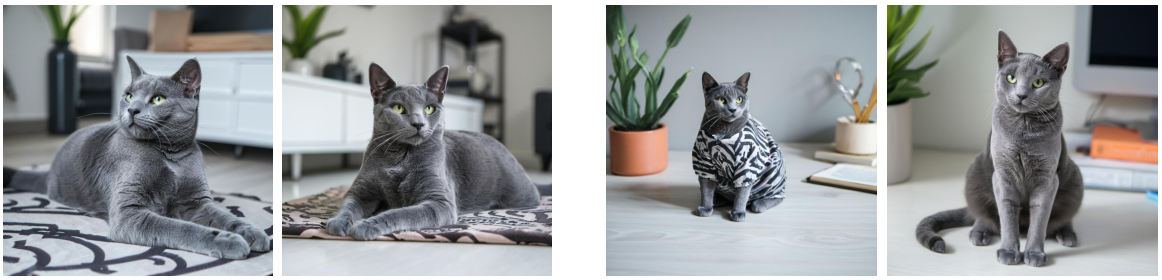
FLoRA (Si et al., 2024) introduces an extra core based on Tucker decomposition to maintain the consistent topological structure with the original space. The core is quite similar to our mixer. The differences are:

- **motivation**: FLoRA aims to maintain the structural integrity of the involved high-dimensional spaces, while we try to analyze LoRA and two-subspaces-mixing LoRA in the view of subspace.
- **initialization**: FLoRA initializes the core as the zero matrix, while MoSLoRA employs the Kaiming uniform distribution and orthogonal matrix. As shown in Table 4, MoSLoRA outperforms FLoRA on the commonsense reasoning tasks.

E Cases of Generated Images

Figure 7, 8, 9, and 10 show the specific generated images and paired prompts. For the definition images of these subjects, please refer to the official data[8] of DreamBooth.

[8] DreamBooth dataset

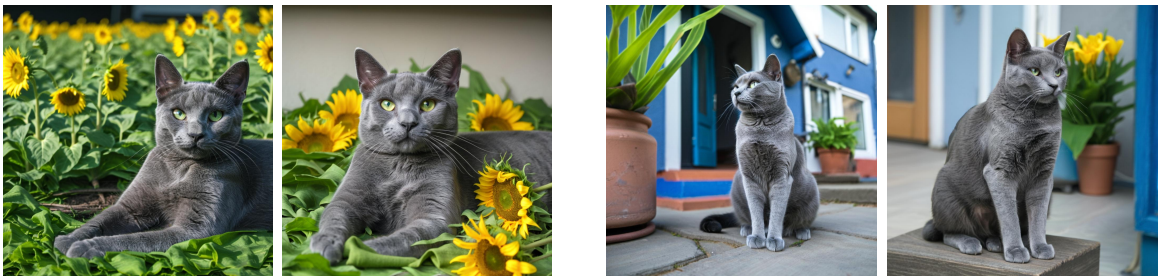


A [V] cat on the top of a white rug

A cube shaped [V] cat

A [V] cat floating on water

A [V] cat on the beach



A [V] cat on the top of green grass with sunflowers around it

A [V] cat with a blue house in the background

A [V] cat with a wheat field in the background

A wet [V] cat

MoSLoRA **LoRA** **MoSLoRA** **LoRA**

Figure 7: Cases of generated images and paired prompts for the subject *cat*.

Figure 8: Cases of generated images and paired prompts for the subject *dog*.

Figure 9: Cases of generated images and paired prompts for the subject *can*.

A [V] grey sloth plushie in the snow

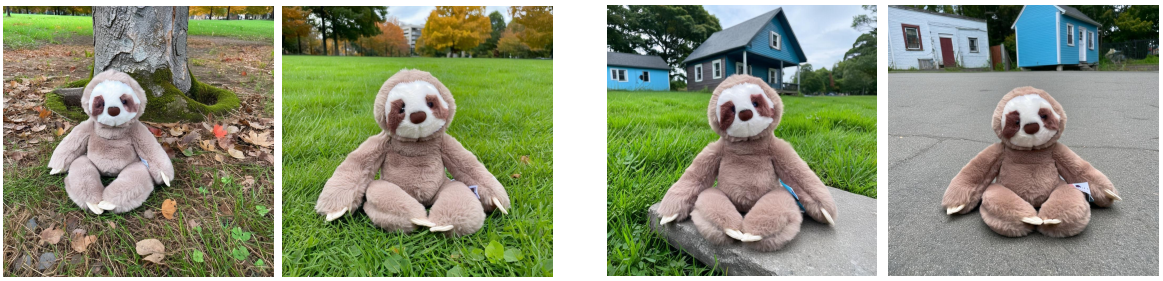
A [V] grey sloth plushie on the beach

A [V] grey sloth plushie on the top of a dirt road

A [V] grey sloth plushie floating on water

A [V] grey sloth plushie with a city in the background

A [V] grey sloth plushie on the top of a purple rug in a forest



A [V] grey sloth plushie with a tree and autumn leaves in the background

A [V] grey sloth plushie with a blue house in the background

MoSLoRA **LoRA** **MoSLoRA** **LoRA**

Figure 10: Cases of generated images and paired prompts for the subject *grey sloth plushie*.