Put the Space of LoRA Initialization to the Extreme to Preserve Pre-trained Knowledge

Pengwei Tang[1,2,3], **Xiaolin Hu**[4 *], **Yong Liu**[1,2,3],
Lizhong Ding[5], **Dongjie Zhang**[6], **Xing Wu**[6,7], **Debing Zhang** [6]

[1]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China,
[2] Beijing Key Laboratory of Research on Large Models and Intelligent Governance,
[3]Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE,
[4]Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University,
[5]Beijing Institute of Technology,
[6]Xiaohongshu Inc.,
[7]University of Chinese Academy of Sciences, Beijing, China
tangpwei@ruc.edu.cn, xiaolinhu@xmu.edu.cn

Abstract

Low-Rank Adaptation (LoRA) is the leading parameter-efficient fine-tuning method for Large Language Models (LLMs), but it still suffers from catastrophic forgetting. Recent work has shown that specialized LoRA initialization can alleviate catastrophic forgetting. There are currently two approaches to LoRA initialization aimed at preventing knowledge forgetting during fine-tuning: (1) making residual weights close to pre-trained weights, and (2) ensuring the space of LoRA initialization is orthogonal to pre-trained knowledge. The former is what current methods strive to achieve, while the importance of the latter is not sufficiently recognized. We find that the space of LoRA initialization is the key to preserving pre-trained knowledge rather than the residual weights. Existing methods like MiLoRA propose making the LoRA initialization space orthogonal to pre-trained weights. However, MiLoRA utilizes the null space of pre-trained weights. Compared to pre-trained weights, the input activations of pre-trained knowledge take into account the parameters of all previous layers as well as the input data, while pre-trained weights only contain information from the current layer. Moreover, we find that the effective ranks of input activations are much smaller than those of pre-trained weights. Thus, the null space of activations is more accurate and contains less pre-trained knowledge information compared to that of weights. Based on these, we introduce LoRA-Null, our proposed method that initializes LoRA in the null space of activations. Experimental results show that LoRA-Null effectively preserves the pre-trained world knowledge of LLMs while achieving good fine-tuning performance, as evidenced by extensive experiments.

Code — https://github.com/HungerPWAY/LoRA-Null
Extended version — https://arxiv.org/abs/2503.02659

Introduction

Low-rank adaptation (LoRA), which has already become the leading parameter-efficient fine-tuning (PEFT) (Lester, Al-Rfou, and Constant 2021; Li and Liang 2021; Houlsby et al. 2019; Hu et al. 2022; Tang, Hu, and Liu 2025; Yao et al. 2025) method for Large Language Models (LLMs), assumes weight changes during fine-tuning have a low-rank structure (Hu et al. 2022). To fine-tune pre-trained weight $\mathbf{W}_0 \in \mathbb{R}^{n\times m}$, LoRA uses two low-rank matrices $\mathbf{A} \in \mathbb{R}^{r\times m}$ and $\mathbf{B} \in \mathbb{R}^{n\times r}$ with $r \ll \min(m, n)$, to represent the corresponding weight change, *i.e.*, $\mathbf{\Delta W} = \mathbf{BA}$. The vanilla LoRA uses random Gaussian to initialize the down-projection matrix $\mathbf{A}$ and zero to initialize the up-projection matrix $\mathbf{B}$, which makes $\mathbf{BA} = \mathbf{0}$ at the beginning of fine-tuning. LoRA reduces trainable parameters by tuning only low-rank matrices.

Method	Making $\mathbf{W}_0'$ close to $\mathbf{W}_0$	$\mathbf{BA}$ orthogonal to Pretraind Knowlegde
LoRA	✓	X
MiLoRA	✓	✓
CorDA	✓	X
LoRA-Null	X	✓

Table 1: Comparison of methods in residual weights and LoRA initialization space.

Recent studies (Biderman et al. 2024) show that LoRA suffers less from this issue compared to full fine-tuning, but it still exhibits significant catastrophic forgetting (Biderman et al. 2024; Dou et al. 2024; Wu et al. 2024). For LoRA fine-tuning, catastrophic forgetting can be mitigated through appropriate initialization techniques, such as CorDA (Yang et al. 2024) and MiLoRA (Wang et al. 2025a). These LoRA initialization methods use not zero and not random Gaussian initialization to initialize $\mathbf{A}$ and $\mathbf{B}$. Unlike vanilla LoRA, they require the base weights to be adjusted such that the initial merged weights $\mathbf{W}_0 + \mathbf{BA}$ equal the original weights. The adjusted base weights, referred to as the *residual weights*, are denoted by $\mathbf{W}_0' = \mathbf{W}_0 - \mathbf{BA}$.

MiLoRA (Wang et al. 2025a) updates the minor singular components of the pre-trained weights, preserving pre-trained knowledge by freezing the principal components and using the less-optimized subspace for learning. CorDA (Yang

*Corresponding Author.

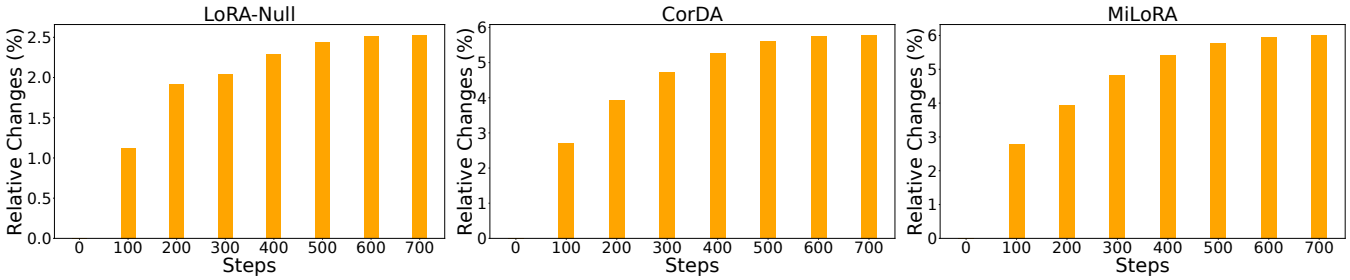


Figure 1: The relative changes of LoRA-Null, CorDA and MiLoRA. It is caculated by $\|\mathbf{A}^* - \mathbf{A}_0\|_F / \|\mathbf{A}_0\|_F$, where $\mathbf{A}^*$ is the tuned parameters. We calculate the relative changes on the $\mathbf{A}$ matrix of the key projection of layer 0 on LLaMA-3.2-3B.

et al. 2024) conducts Singular Value Decomposition (SVD) (Strang 2022) on pre-trained weights guided by the covariance matrix of pre-trained knowledge activations, integrating the context into the principal components to keep knowledge.

Based on the observations from MiLoRA and CorDA, we can summarize that there are currently two approaches to LoRA initialization aimed at preventing knowledge forgetting during fine-tuning: (1) making residual weights close to pre-trained weights, which are adopted by both CorDA and MiLoRA; (2) ensuring the space of LoRA initialization is orthogonal to pre-trained knowledge, which are used by MiLoRA. In Table 1, we show that comparison of methods in residual weights and the space of LoRA initializations. Because it is believed that LoRA adapters $\mathbf{BA}$ will change during fine-tuning, keeping the frozen residual weights close to the pre-trained weights seems like the key factor that enables the preservation of pre-trained knowledge. However, for CorDA and MiLoRA, the LoRA initialization is the component of the pre-trained weights (Yang et al. 2024; Wang et al. 2025a). Just as proper fine-tuning keeps weights close to pre-trained weights, fine-tuned LoRA adapters remain close to their initializations. In Figure 1, we show that the relative changes of LoRA adapters are small. This suggests that, for knowledge preservation, focusing the primary objective on residual weights may not be that important. **A supporting example is that vanilla LoRA, despite freezing the full pre-trained weights, does not outperform MiLoRA or CorDA in knowledge preservation** (Yang et al. 2024). This motivates us to turn our attention toward the approach of ensuring the space of adapters initialization is orthogonal to pre-trained knowledge.

From the perspective of making the space of LoRA initialization orthogonal to pre-trained knowledge, MiLoRA has done this by making the space of LoRA initialization orthogonal to $\mathbf{W}_0$. Given that the nonlinear neural network unit is $\sigma(\mathbf{W}\mathbf{x})$, we can also consider the null space of $\mathbf{X}_{\text{pre}}$. $\mathbf{X}_{\text{pre}}$ contains information from the parameters of all previous layers and the input data, whereas $\mathbf{W}_0$ only considers the current layer. This suggests that we should choose to analyze $\mathbf{X}_{\text{pre}}$. Meanwhile, we also show that the effective rank of $\mathbf{X}_{\text{pre}}$ is much smaller than that of $\mathbf{W}_0$. This means that a higher proportion of information in $\mathbf{X}_{\text{pre}}$ is concentrated in the subspace corresponding to the highest singular values, while the subspace associated with the smallest singular values contains a lower proportion of pre-trained knowledge.

In this paper, we propose LoRA-Null, *i.e.* Low-Rank Adaptation via Null Space of pre-trained knowledge activations, to address catastrophic forgetting. LoRA-Null randomly samples a small amount of data from datasets that are representative of the pre-training knowledge to get the activation $\mathbf{X}_{\text{pre}}$. Then, we extract the null space of $\mathbf{X}_{\text{pre}}$. Following Meng, Wang, and Zhang; Wang et al.; Yang et al., we also let the component of $\mathbf{W}_0$ be the LoRA initialization. Thus, we make the LoRA initialization $\mathbf{BA} = \mathbf{W}_0\mathbf{U}_{\text{null}}\mathbf{U}_{\text{null}}^\top$.

Our contributions can be concluded as follows:

- We find that making the space of LoRA initialization orthogonal to pre-trained knowledge is more critical for preserving pre-trained knowledge than making the residual weights close to the pre-trained weights.
- We find that the effective rank of the input activation is much smaller than that of the pre-trained weights. To push orthogonal LoRA initialization to the extreme for knowledge preservation, i.e., Low-Rank Adaptation via Null Space, which constructs adapters initialized from the null space of the pre-trained knowledge activation.
- We conduct extensive experiments on three tasks, showing that our proposed LoRA-Null achieves good downstream performance while preserving pre-trained knowledge.

Related Work

Initialization methods for LoRA. The vanilla LoRA uses a random Gaussian distribution to initialize the up-projection matrix $\mathbf{A}$ and uses a zero matrix to initialize the down-projection matrix $\mathbf{B}$. Recent studies have shown that the proper initialization can improve the performance of LoRA (Meng, Wang, and Zhang 2024; Yang et al. 2024; Wang et al. 2025a). LoRA initialization methods can be broadly classified into two main categories: (1) those designed to accelerate downstream task performance, exemplified by PiSSA (Meng, Wang, and Zhang 2024), and the instruction-previewed mode of CorDA (Yang et al. 2024); and (2) those aimed at preserving pre-trained knowledge, such as MiLoRA (Wang et al. 2025a) and the knowledge-preserved mode of CorDA (Yang et al. 2024). PiSSA adopts principal singular values and singular vectors of the pre-trained weights as the initialization for LoRA adapters (Meng, Wang, and Zhang 2024). To extract input-relevant pre-trained

weight components, CorDA performs SVD on the product of pre-trained weights and input activation covariance matrices. The top singular values/vectors are then multiplied by the inverse of the input activation's covariance matrix (Yang et al. 2024). CorDA operates in two distinct modes: an instruction-previewed mode, which utilizes input activations from the downstream task, and a knowledge-preserved mode, where input activations are sampled from the pre-training task. MiLoRA adopts minor singular values and singular vectors of the pre-trained weights as the initialization for LoRA adapters (Wang et al. 2025a). In this paper, we focus on LoRA initialization methods designed for knowledge preservation and further elucidate the mechanisms of LoRA initialization in achieving knowledge preservation.

Activation-aware knowledge preservation. The activation is the output of a neuron after its weighted sum of inputs has been transformed by a non-linear activation function. Recent studies have shown that considering activation features can lead to better pre-trained knowledge preservation compared to only focusing on pre-trained weights (Lin et al. 2024; Wang et al. 2025b; Yang et al. 2024). The activation-aware knowledge preservation is widely used in model compression and LoRA initialization. Activation-aware Weight Quantization (AWQ) is a model quantization method that identifies and scales a small percentage of crucial weights in LLMs based on activation distributions, significantly reducing quantization error and accelerating inference (Lin et al. 2024). SVD-LLM is an SVD-based post-training LLM compression method that uses truncation-aware data whitening derived from activations and sequential low-rank approximation to recover accuracy (Wang et al. 2025b). CorDA extracts data-relevant pre-trained weight components by applying SVD to the product of pre-trained weights and input activation covariance matrices (Yang et al. 2024). The dataset sampled to construct the input activation is called the calibration set. To retain pre-training knowledge, these data are sampled from dataset that embodies the pre-training capabilities.

Preliminaries

LoRA

For pretrained weights $\mathbf{W}_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, LoRA introduces a down-projection matrix $\mathbf{A} \in \mathbb{R}^{r \times d_{\text{in}}}$ and an up-projection matrix $\mathbf{B} \in \mathbb{R}^{d_{\text{out}} \times r}$, where $r \ll \min(d_{\text{out}}, d_{\text{in}})$, which is

$$\mathbf{y} = \mathbf{W}^*\mathbf{x} = \mathbf{W}_0\mathbf{x} + \Delta\mathbf{W}\mathbf{x} = \mathbf{W}_0\mathbf{x} + \mathbf{B}\mathbf{A}\mathbf{x}, \quad (1)$$

where $\mathbf{W}^*$ is the fine-tuned weights, $\Delta\mathbf{W}$ is the change in weights, $\mathbf{x} \in \mathbb{R}^{d_{\text{in}}}$ is the input, and $\mathbf{y} \in \mathbb{R}^{d_{\text{out}}}$ is the output.

Input Activations of Pre-trained Knowledge: $\mathbf{X}_{\text{pre}}$

Following Yang et al., we randomly collect some samples from the training data of some tasks that are representative of the pre-trained knowledge. Here, the collected samples are used to compute their corresponding input activations. Denote $\mathbf{X}_{\text{pre}} \in \mathcal{R}^{d_{\text{in}} \times BL}$ as the input activation matrix of a linear layer derived from these sampled data, where d_{in} is the input dimension, B is the number of collected samples, and L represents the maximum sequence length. **Note that $\mathbf{X}_{\text{pre}}$ denotes the input activations of pre-trained knowledge, while $\mathbf{x}$ denotes any input**.

MiLoRA and CorDA

MiLoRA (Wang et al. 2025a) only updates the minor singular components of the weight matrix while freezing the principal singular components. Let $\texttt{SVD}(\mathbf{W}_0) = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^\top$ and $R = \text{rank}(\mathbf{W}_0)$, MiLoRA uses $\hat{\mathbf{U}}_{[:,:R-r]}\hat{\mathbf{\Sigma}}_{[:R-r]}\hat{\mathbf{V}}^\top_{[:R-r,:]}$ as the residual weight and use $\hat{\mathbf{U}}_{[,-r:]}\hat{\mathbf{\Sigma}}_{[-r:]}\hat{\mathbf{V}}^\top_{[-r:,:]}$ as the LoRA initialization. CorDA (Yang et al. 2024) is a LoRA initialization method that uses data from downstream tasks or pre-trained knowledge. CorDA uses the covariance matrix $\mathbf{C} = \mathbf{X}_{\text{pre}}\mathbf{X}^\top_{\text{pre}}$ to extract the components of the pre-trained weight matrices that are the most related to the provided data. CorDA first uses $\texttt{SVD}(\mathbf{W}_0\mathbf{C}) = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^\top$. For the knowledge-preserved adaptation of CorDA, CorDA uses $\hat{\mathbf{U}}_{[:,:R-r]}\hat{\mathbf{\Sigma}}_{[:R-r]}\hat{\mathbf{V}}^\top_{[:R-r,:]}\mathbf{C}^{-1}$ as the residual weights and uses $\hat{\mathbf{U}}_{[:,-r:]}\hat{\mathbf{\Sigma}}_{[-r:]}\hat{\mathbf{V}}^\top_{[-r:,:]}\mathbf{C}^{-1}$.

Effective Rank

Given a non-zero matrix $\mathbf{K} \in \mathbb{R}^{M \times N}$ with singular values $\sigma_1, \sigma_2, \ldots, \sigma_Q$ where $Q = \min\{M, N\}$, we define the normalized singular value distribution:

$$p_i = \frac{\sigma_i}{\sum_{j=1}^{Q} \sigma_j}, \quad i = 1, \ldots, Q, \quad (2)$$

where p_i represents the relative weight of the i-th singular value in the spectrum. The **effective rank** is defined as

$$\text{eRank}(\mathbf{K}) = \exp\left(-\sum_{i=1}^{Q} p_i \log p_i\right), \quad (3)$$

where it quantifies the spectral entropy of singular values, giving an information-theoretic characterization of a matrix's effective rank (Roy and Vetterli 2007). A high effective rank indicates that a matrix's singular values are relatively evenly distributed, while a low effective rank indicates that most information is concentrated in the top few singular values.

Motivation

The analysis of CorDA and MiLoRA from the perspective of $\mathbf{W}'_0$

In this section, we first analyze the CorDA and MiLoRA from the perspective of $\mathbf{W}'_0$. We present two theorems as follows, where the proofs are given in extended version.

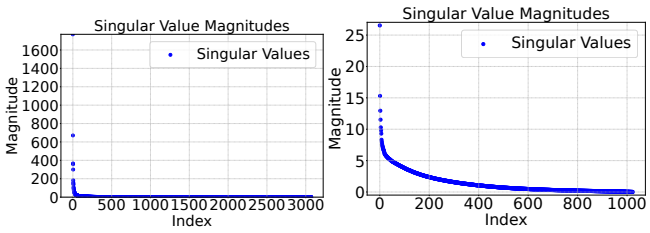
Theorem 1. *The initialization of MiLoRA is the solution of*

$$\min_{\mathbf{W}'_0} \|\mathbf{W}'_0 - \mathbf{W}_0\|_{\text{F}}, \textit{s.t. rank}\,(\mathbf{W}'_0) = R - r. \quad (4)$$

Theorem 2. *The initialization of CorDA is the solution of*

$$\min_{\mathbf{W}'_0} \|\mathbf{W}'_0\mathbf{X}_{pre} - \mathbf{W}_0\mathbf{X}_{pre}\|_{\text{F}}, \textit{s.t. rank}\,(\mathbf{W}'_0) = R - r. \quad (5)$$

We can observe that both MiLoRA and CorDA aim to make $\mathbf{W}'_0$ close to $\mathbf{W}_0$. However, in Figure 1, we have shown that the contribution of $\mathbf{BA}$ after fine-tuning to the pre-trained knowledge is non-negligible. Meanwhile, we note that although LoRA freezes the original weights $\mathbf{W}_0$, it does not preserve the pre-trained knowledge well. Therefore, it is not so important to make $\mathbf{W}'_0$ close to $\mathbf{W}_0$.



(a) $\mathbf{X}_{\text{pre}}$ (b) $\mathbf{W}_0$

Figure 2: The singular value magnitudes of $\mathbf{X}_{\text{pre}}$ and $\mathbf{W}_0$ on the key matrix of layer 0 on LLaMA-3.2-3B

	qproj	kproj	vproj	upproj	downproj
$\mathbf{W}_0$ (0)	1214.11	548.30	892.36	2871.72	2883.30
$\mathbf{X}_{\text{pre}}$ (0)	101.28	101.28	101.28	553.59	758.06
$\mathbf{W}_0$ (1)	1788.42	714.95	941.63	2858.23	2882.98
$\mathbf{X}_{\text{pre}}$ (1)	85.63	85.63	85.63	657.79	1.27
$\mathbf{W}_0$ (27)	2089.48	851.92	943.69	2757.13	2838.64
$\mathbf{X}_{\text{pre}}$ (27)	194.22	194.22	194.22	302.78	79.97

Table 2: The effective rank of $\mathbf{X}_{\text{pre}}$ and $\mathbf{W}_0$ on LLaMA-3.2-3B. The number in "()" indicates the layer index.

The analysis of CorDA and MiLoRA from the perspective of the LoRA initialization space

From the perspective of the LoRA initialization space, we observe that the initialization space of MiLoRA is the null space of $\mathbf{W}_0$, whereas the initialization space of CorDA is neither the null space of $\mathbf{W}_0$ nor that of $\mathbf{X}_{\text{pre}}$.

When the LoRA initialization space contains the principal components of pre-trained knowledge, since the LoRA matrices are trainable, LoRA may drastically amplify the components associated with pre-trained knowledge; moreover, these principal components, when perturbed, can more easily lead to the corruption of the pre-trained knowledge. **Therefore, it is the key for knowledge preserving to make $\mathbf{BA}$ orthogonal to pre-trained knowledge.**

Comparing the null space of $\mathbf{X}_{\text{pre}}$ and $\mathbf{W}_0$

The output of a neural network is determined by a series of nonlinear units $\sigma(\mathbf{W}\mathbf{x})$. Thus, for the null space for LoRA initialization, we can take into account either $\mathbf{W}_0$ or $\mathbf{X}_{\text{pre}}$. Recent studies have shown that $\mathbf{X}_{\text{pre}}$ can reflect the characteristics of pre-trained data (Lin et al. 2024; Wang et al. 2025b; Yang et al. 2024). Next, we will explain why the null space of $\mathbf{X}_{\text{pre}}$ is better than that of $\mathbf{W}_0$, from two perspectives.

First, $\mathbf{W}_0$ has weaker importance of information contained in pre-training tasks than $\mathbf{X}_{\text{pre}}$. $\mathbf{X}_{\text{pre}}$ takes into account the parameters from all previous layers as well as the input data, whereas $\mathbf{W}_0$ only reflects its own parameters.

Second, we analyze $\mathbf{X}_{\text{pre}}$ and $\mathbf{W}_0$ from the perspective of effective rank. The effective rank is calculated by Eq.3. In Figure 2, we can observe that the singular value magnitudes of $\mathbf{X}_{\text{pre}}$ are steeper than those of $\mathbf{W}_0$. This means that the information contained in $\mathbf{X}_{\text{pre}}$ is more concentrated in the principal subspace compared to $\mathbf{W}_0$. From the perspective of effective rank, this implies that the effective rank of $\mathbf{X}_{\text{pre}}$ is much smaller than that of $\mathbf{W}_0$, as is demonstrated in Table 2. Actually, $\mathbf{X}_{\text{pre}}$ contains information about all the parameters and input data from previous layers, which is the reason for its smaller effective rank. This is because the more precise the information in $\mathbf{X}_{\text{pre}}$ is, the lower the uncertainty of the information it contains, and thus the lower the effective rank will be. From the perspective of effective rank alone, the null space of $\mathbf{X}_{\text{pre}}$ contains less pre-trained knowledge compared to that of $\mathbf{W}_0$. Therefore, initializing LoRA in the null space of $\mathbf{X}_{\text{pre}}$ will achieve better knowledge preservation than initializing it in the null space of $\mathbf{W}_0$.

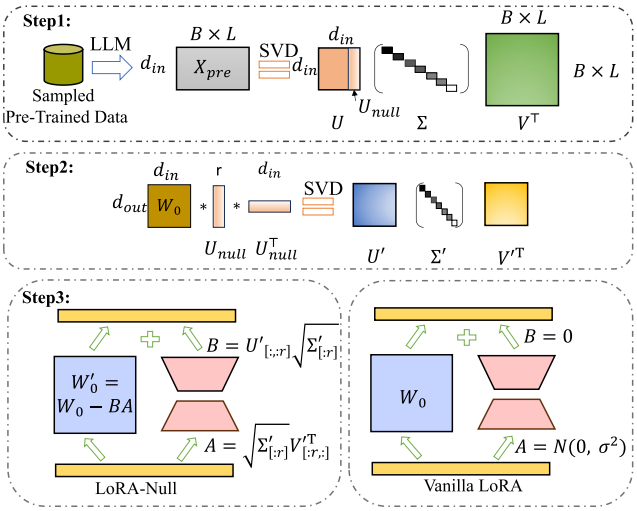


Figure 3: An illustration of LoRA-Null. We first use SVD on the pre-trained knowledge activations to obtain the $\mathbf{U}_{\text{null}}$. Then, we use $\mathbf{W}_0(\mathbf{U}_{\text{null}}\mathbf{U}_{\text{null}}^\top)$ to extract the projection of $\mathbf{W}_0$ onto the null space of $\mathbf{X}_{\text{pre}}$. Finally, we conduct SVD on $\mathbf{W}_0(\mathbf{U}_{\text{null}}\mathbf{U}_{\text{null}}^\top)$ to initialize $\mathbf{A}$ and $\mathbf{B}$ and replace $\mathbf{W}_0$ with $\mathbf{W}_0 - \mathbf{BA}$. **LoRA-Null** only allows updating $\mathbf{A}$ and $\mathbf{B}$.

Our Method

In this section, we introduce our proposed LoRA-Null. We let $\mathbf{BA}$ in the null space of $\mathbf{X}_{\text{pre}}$ and do not consider the residual weight $\mathbf{W}_0'$. Following MiLoRA and CorDA (Yang et al. 2024; Wang et al. 2025a), we let $\mathbf{BA}$ is the component of $\mathbf{W}_0$. Thus, we let $\mathbf{BA} = \mathbf{W}_0\mathbf{U}_{\text{null}}\mathbf{U}_{\text{null}}^\top$, where $\mathbf{U}_{\text{null}}$ is the left null space of $\mathbf{X}_{\text{pre}}$. $\mathbf{BA} = \mathbf{W}_0\mathbf{U}_{\text{null}}\mathbf{U}_{\text{null}}^\top$ means $\mathbf{BA}$ is the projection of $\mathbf{W}_0$ onto the null space of $\mathbf{X}_{\text{pre}}$. In contrast, the LoRA initialization of MiLoRA is the projection of $\mathbf{W}_0$ onto the null space of $\mathbf{W}_0$.

To get the left null space of $\mathbf{X}_{\text{pre}}$, we perform SVD of $\mathbf{X}_{\text{pre}}$:

$$\mathbf{X}_{\text{pre}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \sum_{i=1}^{R} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top, \tag{6}$$

where $\mathbf{U} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ and $\mathbf{V} \in \mathbb{R}^{d_{B\times L} \times d_{B\times L}}$ are orthogonal matrices, and $\mathbf{\Sigma} \in \mathbb{R}^{d_{\text{in}} \times d_{B\times L}}$ is a diagonal matrix

(a) Math on LLaMA-2-7B

Method	**#Para**	Trivia QA	NQ open	WebQS	Avg1	Avg1(Per)	GSM8k	Math	Avg2	GM
LLaMA-2-7B	-	52.51	18.83	5.91	25.75	100.00	-	-	-	-
LoRA	320M	45.95	1.16	**6.64**	17.91	64.56	42.99	6.26	24.63	21.00
PiSSA	320M	43.70	2.30	6.50	17.50	65.15	**51.70**	7.64	**29.67**	22.79
MiLoRA	320M	47.02	3.66	6.10	18.93	69.66	41.47	6.20	23.84	21.24
CorDA	320M	48.99	7.15	5.76	20.63	76.24	41.47	8.22	24.85	22.64
LoRA-Null	320M	**50.02**	**7.98**	6.55	**21.52**	**79.21**	44.43	**8.80**	26.62	**23.93**

(b) Code on LLaMA-2-7B

Method	**#Para**	Trivia QA	NQ open	WebQS	Avg1	Avg1(per)	HumanEval	MBPP	Avg2	GM
LLaMA-2-7B	-	52.51	18.83	5.91	25.75	100.00	-	-	-	-
LoRA	320M	**51.30**	12.24	8.66	24.07	87.57	17.19	21.29	19.24	21.52
PiSSA	320M	46.08	14.40	**9.25**	23.24	84.25	**20.61**	**23.84**	**22.23**	**22.73**
MiLoRA	320M	49.33	13.49	7.19	23.34	88.53	18.36	20.97	19.67	21.43
CorDA	320M	49.01	15.24	6.59	23.61	91.42	18.36	21.65	20.01	21.74
LoRA-Null	320M	51.29	**16.51**	7.87	**25.22**	**95.12**	18.36	21.65	20.01	22.46

(c) Instruction Following on LLaMA-2-7B

Method	**#Param**	Trivia QA	NQ open	WebQS	Avg1	Avg1 (Per)	MTBench	GM
LLaMA-2-7B	-	52.51	18.83	5.91	25.75	100.00	-	-
LoRA	320M	46.06	9.72	**7.28**	21.02	79.78	3.81	8.95
PiSSA	320M	35.76	10.17	5.61	17.18	72.34	**4.21**	8.50
MiLoRA	320M	45.98	11.94	6.74	21.55	83.66	3.79	9.04
CorDA	320M	47.10	13.63	6.74	22.49	87.36	3.89	9.35
LoRA-Null	320M	**48.09**	**14.71**	6.69	**23.16**	**89.90**	4.02	**9.65**

(d) Math on LLaMA3.2-3B

Method	**#Para**	Trivia QA	NQ open	WebQS	Avg1	Avg1(Per)	GSM8k	Math	Avg2	GM
LLaMA-3.2-3B	-	50.77	13.55	9.25	24.52	100.00	-	-	-	-
LoRA	195M	43.62	8.28	8.02	19.97	77.91	55.65	12.72	34.19	26.13
PiSSA	195M	42.88	7.26	8.91	19.68	78.12	**63.84**	**15.7**	**39.77**	27.98
MiLoRA	195M	46.26	10.97	8.17	21.80	86.80	56.56	13.58	35.07	27.65
CorDA	195M	46.77	10.22	**9.20**	22.06	89.00	58.91	14.70	36.81	28.50
LoRA-Null	195M	**49.03**	**11.52**	9.01	**23.19**	**93.18**	58.76	14.06	36.41	**29.06**

Table 3: Results of LoRA-Null vs. baselines on LLaMA-2-7B (LLaMA-3.2-3B). The first row shows the pre-trained performance. **Bold** indicates the best result; underlined indicates the runner-up. "Avg1" is the average preservation score, "Avg1(Per)" is the percentage of preserved performance relative to pre-trained, "Avg2" is the average of downstream task scores, and "GM" is the geometric mean of Avg1 and Avg2.

with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_R \geq 0$ (with $R = \min(d_{\text{in}}, B \times L) = d_{\text{in}}$, $B \times L \gg d_{\text{in}}$). Given a predefined LoRA rank r (typically $r \ll d_{\text{in}}$), we approximate the left null space of $\mathbf{X}_{\text{pre}}$ by selecting the r left singular vectors associated with the *smallest* singular values. Formally, if $\mathbf{U} = [\mathbf{U}_1|\mathbf{U}_2]$ with $\mathbf{U}_2 \in \mathbb{R}^{d_{\text{in}} \times r}$, we treat $\mathbf{U}_2$ as the approximate null space under the assumption that the trailing singular values $\{\sigma_{d_{\text{in}}-r+1}, \ldots, \sigma_{d_{\text{in}}}\}$ are negligibly small, as evidenced in Figure 2. This leads to the approximation:

$$\mathbf{U}_2^\top \mathbf{X}_{\text{pre}} \approx \mathbf{0}. \tag{7}$$

Hereafter, we use $\mathbf{U}_{\text{null}}$ to represent $\mathbf{U}_2$.

As shown in Figure 3, we perform SVD on the pre-trained weights projected onto the null space, which is:

$$\mathbf{W}_0 \mathbf{U}_{\text{null}} \mathbf{U}_{\text{null}}^\top = \mathbf{U}' \mathbf{\Sigma}' \left(\mathbf{V}'\right)^\top, \tag{8}$$

where $\mathbf{U}'$ and $\mathbf{V}'$ are orthogonal matrices of left and right singular vectors, respectively, and $\mathbf{\Sigma}'$ is a diagonal matrix of singular values. And we can observe that

$$\text{rank}\left(\mathbf{W}_0 \mathbf{U}_{\text{null}} \mathbf{U}_{\text{null}}^\top\right) \leq \text{rank}\left(\mathbf{U}_{\text{null}}\right) = r. \tag{9}$$

We then initialize the adapter matrices $\mathbf{B}$ and $\mathbf{A}$ as:

$$\mathbf{B} = \mathbf{U}'_{[:,:r]} \sqrt{\mathbf{\Sigma}'_{[:r]}}, \mathbf{A} = \sqrt{\mathbf{\Sigma}'_{[:r]}} \mathbf{V}'^\top_{[:r,:]}. \tag{10}$$

We prove that the columns of $\mathbf{A}^\top$ lie in the column space of $\mathbf{U}_{\text{null}}$ in extended version. The residual weight matrix is

$$\mathbf{W}_0' = \mathbf{W}_0 - \mathbf{BA}. \tag{11}$$

During fine-tuning, only $\mathbf{A}$ and $\mathbf{B}$ are updated while $\mathbf{W'}_0$ are frozen. This is our proposed LoRA-Null.

Theorem 3. *The initialization of LoRA-Null is not the solution of Eq. 4 and Eq. 5.*

The proof of Theorem 3 and the corresponding experiments are provided in extended version. Theorem 3 shows that LoRA-Null only considers the space of LoRA initialization and does not consider making $\mathbf{W}_0'$ close to $\mathbf{W}_0$. This is the origin of our title: we push the idea of aligning the space of LoRA initialization to be orthogonal to pre-trained knowledge to the extreme in order to preserve pre-trained knowledge without considering making $\mathbf{W}_0'$ close to $\mathbf{W}_0$.

Experiments

Experimental Setup

Models and Datasets. We fine-tune the pre-trained LLMs—**LLaMA-2-7B** and **LLaMA-3.2-3B**—on **Math**, **Code**, and **Instruction Following** tasks. Following Yang et al., the pre-trained knowledge is assessed using exact match scores (%) on the **TriviaQA** (Joshi et al. 2017), **NQ Open** (Lee, Chang, and Toutanova 2019), and **WebQS** (Berant et al. 2013) datasets. For Math, LLMs are trained on MetaMathQA (Yu et al. 2024) and tested on the **GSM8k** (Cobbe et al. 2021) and **Math** (Yu et al. 2024) validation sets. For Code, LLMs are trained on CodeFeedback (Wei et al. 2024) and tested on the **Humaneval** (Chen et al. 2021) and **MBPP** (Austin et al. 2021). For Instruction Following, LLMs are trained on WizardLM-Evol-Instruct (Xu et al. 2024) and tested on the **MTBench** (Zheng et al. 2023).

Implementation Details. We follow the same training configuration as (Meng, Wang, and Zhang 2024; Yang et al. 2024). Specifically, the optimization process utilizes the AdamW (Loshchilov and Hutter 2019) optimizer with a batch size of 128 and a learning rate of 2×10^{-5}. We use cosine annealing schedules alongside a warmup ratio of 0.03. The training is conducted exclusively on the initial 100,000 conversations from the dataset for one epoch, where the loss calculation is based solely on the response. The rank of LoRA is set to 128. Our experiments are carried out on a single NVIDIA H800 80GB GPU. Following (Yang et al. 2024), we randomly sample 256 data points from NQ Open with a maximum length of 1024 as the calibration set, which is used to construct the input activations representing the pre-trained knowledge.

Baselines. The baselines are: (1) LoRA (Hu et al. 2022); (2) PiSSA (Meng, Wang, and Zhang 2024); (3) MiLoRA (Wang et al. 2025a); and (4) CorDA (Yang et al. 2024).

Main results on LLaMA-2-7B and LLaMA-3.2-3B

Tables 3a, 3b and 3c and 3d present our experimental results on LLaMA-2-7B and LLaMA-3.2-3B across Math, Code and Instruction Following tasks, which demonstrate the superior knowledge preservation of our proposed LoRA-Null, as well as its relatively good performance on downstream tasks. We observe that LoRA-Null achieves the best knowledge preservation performance on Trivia QA and NQ Open compared to the baselines, except in the code for Trivia QA, where LoRA achieves 50.30 and LoRA-Null attains 50.29. For WebQS, LoRA-Null achieves the second-best overall performance. Meanwhile, we can see that LoRA-Null achieves the best average performance in knowledge preservation across the board. In terms of performance relative to the pre-trained models, LoRA-Null achieves an average improvement of 3.35% in knowledge preservation percentage "Avg(Per)" compared to CorDA. For downstream tasks, LoRA-Null is second only to PiSSA in overall performance and achieves the best results on Math on LLaMA-2-7B. Although LoRA-Null underperforms CorDA slightly on downstream tasks with LLaMA-3.2-3B, our hyperparameters follow those of CorDA. As shown in Table 4a, when the sample size of the calibration sets is 1024, LoRA-Null outperforms CorDA in both knowledge preservation and fine-tuning performance.

Deep Discussions of LoRA initialization space

We further study the space of the LoRA initialization for LoRA-Null, MiLoRA and CorDA. We analyze the components of the down-projection matrices $\mathbf{A}$ on the subspace of $\mathbf{U}$ in Equation 6 ($\mathbf{U}$ is the left singular matrix of $\mathbf{X}_{\text{pre}}$), as shown in Figure 4. We define the projection matrix as:

$$\mathbf{P} = \mathbf{AU} \in \mathbb{R}^{r \times d_{\text{in}}}. \tag{12}$$

The absolute value sum over columns is computed for each row of $\mathbf{P}$, resulting in a projection absolute value sum vector $\mathbf{pa} \in \mathbb{R}^{d_{\text{in}}}$, where the i-th element is given by:

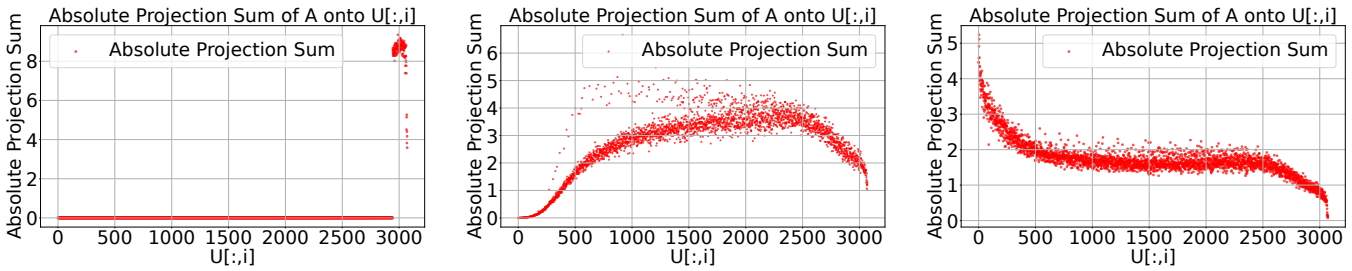
$$\mathbf{pa}_i = \sum_{j=1}^{r} |\mathbf{P}_{j,i}|, \quad i = 1, 2, \ldots, d_{\text{in}}. \tag{13}$$

The horizontal axis is $\mathbf{U}_{[:,i]}$, while the vertical axis is $\mathbf{pa}_i$. A higher value of $\mathbf{pa}_i$ indicates a larger component of the down-projection matrix $\mathbf{A}$ onto the subspace of $\mathbf{U}_{[:,i]}$.

We find that for both CorDA and MiLoRA, their $\mathbf{A}$ do not only lie in $\mathbf{U}_{\text{null}} = \mathbf{U}_{[:,-r:]}$, while the matrix $\mathbf{A}$ of LoRA-Null only lie in $\mathbf{U}_{\text{null}}$. Moreover, the $\mathbf{A}$ in CorDA focuses more on the minor singular vectors of $\mathbf{U}$ compared to MiLoRA. Meanwhile, for CorDA, from a mathematical perspective, $\mathbf{W}_0\mathbf{X}_{\text{pre}}\mathbf{X}_{\text{pre}}^\top = \mathbf{W}_0\mathbf{U}\Sigma^2\mathbf{U}^\top$. This amplifies the components of $\mathbf{W}_0$ along the subspaces in $\mathbf{U}$ with singular values greater than 1, while attenuating those with singular values less than 1. As a result, the subspace corresponding to the smallest singular values in the SVD of $\mathbf{W}_0\mathbf{X}_{\text{pre}}\mathbf{X}_{\text{pre}}^\top$ is dominated by the components of $\mathbf{U}$ associated with small singular values. Meanwhile, $\mathbf{C}^{-1} = \mathbf{U}\Sigma^{-2}\mathbf{U}^\top$ does not change the subspace but amplifies the components corresponding to small singular values. This explains why CorDA outperforms MiLoRA from the perspective of space of LoRA initialization. These further confirm that considering the null space of $\mathbf{X}_{\text{pre}}$ is more effective than that of $\mathbf{W}_0$.

Hyperparameter Analysis

In Table 4a, we conduct experiments by varying the number of calibration sets. First, LoRA-Null exhibits greater stability



(a) LoRA-Null (b) CorDA (c) MiLoRA

Figure 4: Component of the down-projection matrix $\mathbf{A}$ onto the subspace of $\mathbf{U}_{[:,i]}$ for the key matrices in LoRA-Null, CorDA and MiLoRA, applied to layer 0 of LLaMA-3.2-3B.

(a) Different number of calibration sets

Method	**#Rank Size, #Samples Size**	Trivia QA	NQ open	WebQS	Avg1	Avg1(Per)	GSM8k	Math	Avg2	GM
LLaMA-3.2-3B	-	50.77	13.55	9.25	24.52	100.00	-	-	-	-
CorDA	128, 64	42.72	10.00	4.53	19.08	68.97	62.32	15.40	38.86	27.23
LoRA-Null	128, 64	48.69	12.33	9.06	23.36	94.95	59.59	13.94	36.77	29.31
CorDA	128, 256	46.77	10.22	9.20	22.06	89.00	58.91	14.70	36.81	28.50
LoRA-Null	128, 256	49.03	11.52	9.01	23.19	93.00	58.76	14.06	36.41	29.06
CorDA	128, 1024	48.46	9.89	9.30	22.55	89.48	58.15	14.50	36.33	28.62
LoRA-Null	128, 1024	48.98	11.14	10.78	23.63	92.90	59.59	14.54	37.07	29.60

(b) Different ranks of LoRA adapters

Method	**#Rank Size, #Samples Size**	Trivia QA	NQ open	WebQS	Avg1	Avg1(Per)	GSM8k	Math	Avg2	GM
LLaMA-3.2-3B	-	50.77	13.55	9.25	24.52	100.00	-	-	-	-
CorDA	64, 256	48.15	10.97	9.35	22.82	91.93	57.09	12.86	34.98	28.25
LoRA-Null	64, 256	49.10	12.02	8.32	23.15	91.79	55.12	13.06	34.09	28.09
CorDA	128, 256	46.77	10.22	9.20	22.06	89.00	58.91	14.70	36.81	28.50
LoRA-Null	128, 256	49.03	11.52	9.01	23.19	93.00	58.76	14.06	36.41	29.06
CorDA	256, 256	45.62	9.58	5.41	20.20	73.01	61.94	15.86	38.90	28.03
LoRA-Null	256, 256	46.58	10.11	8.12	21.60	84.71	62.40	16.26	39.33	29.15

Table 4: LoRA-Null vs. CorDA with varying calibration sizes and ranks on LLaMA-3.2-3B (Math). The "Avg1" is the average performance of knowledge preservation. The "Avg1(Per)" denotes the average percentage of knowledge preservation. The "Avg2" is the average performance of downstream tasks. "GM" is the geometric mean of Avg1 and Avg2.

than CorDA, as indicated by its smaller performance range. Second, when using a small calibration set, CorDA suffers from a significant decline in its ability to preserve pre-trained knowledge, whereas LoRA-Null still has strong capability. Third, compared to CorDA, LoRA-Null has significantly better average performance in preserving the pre-trained knowledge and improving downstream task performance across different numbers of calibration sets.

In Table 4b, we conduct experiments by varying the rank of LoRA. First, LoRA-Null has a better ability to preserve the pre-trained knowledge than CorDA under different ranks. Second, as the rank increases, CorDA's ability to preserve pre-trained knowledge degrades more rapidly compared to LoRA-Null. Third, as the rank increases, LoRA-Null does not only performs better than CorDA in preserving knowledge but also achieves superior performance on downstream tasks.

Conclusion

In this paper, we investigate how LoRA initialization preserves pre-trained knowledge. We find that it is crucial for knowledge preservation to make LoRA initialization orthogonal to the pre-trained knowledge, rather than making the residual weights close to the pre-trained weights. We find that the effective ranks of input activations are much smaller than those of pre-trained weights. We propose LoRA-Null, a low-rank adaptation method initialized in the null space of the activations of the pre-trained knowledge, which helps preserve LLM knowledge during fine-tuning. Extensive experiments show that LoRA-Null can achieve good downstream performance and effective knowledge preservation. We hope our research will provide useful insights for future studies on LoRA initialization.

Acknowledgments

This research was supported by National Key Research and Development Program of China (NO. 2024YFE0203200), National Natural Science Foundation of China (No. 62476277), CCF-ALIMAMA TECH Kangaroo Fund (No. CCF-ALIMAMA OF 2024008), and Huawei-Renmin University joint program on Information Retrieval. We also acknowledge the support provided by the fund for building worldclass universities (disciplines) of Renmin University of China and by the funds from Beijing Key Laboratory of Big Data Management and Analysis Methods, Gaoling School of Artificial Intelligence, Renmin University of China, from Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education, from Intelligent Social Governance Interdisciplinary Platform, Major Innovation & Planning Interdisciplinary Platform for the "DoubleFirst Class" Initiative, Renmin University of China, from Public Policy and Decision-making Research Lab of Renmin University of China, and from Public Computing Cloud, Renmin University of China.

References

Austin, J.; Odena, A.; Nye, M.; Bosma, M.; Michalewski, H.; Dohan, D.; Jiang, E.; Cai, C.; Terry, M.; Le, Q.; and Sutton, C. 2021. Program Synthesis with Large Language Models. arXiv:2108.07732.

Berant, J.; Chou, A.; Frostig, R.; and Liang, P. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In Yarowsky, D.; Baldwin, T.; Korhonen, A.; Livescu, K.; and Bethard, S., eds., *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1533–1544. Seattle, Washington, USA: Association for Computational Linguistics.

Biderman, D.; Portes, J.; Ortiz, J. J. G.; Paul, M.; Greengard, P.; Jennings, C.; King, D.; Havens, S.; Chiley, V.; Frankle, J.; Blakeney, C.; and Cunningham, J. P. 2024. LoRA Learns Less and Forgets Less. *Transactions on Machine Learning Research*. Featured Certification.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. D. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating large language models trained on code.

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

Dou, S.; Zhou, E.; Liu, Y.; Gao, S.; Shen, W.; Xiong, L.; Zhou, Y.; Wang, X.; Xi, Z.; Fan, X.; Pu, S.; Zhu, J.; Zheng, R.; Gui, T.; Zhang, Q.; and Huang, X. 2024. LoRAMoE: Alleviating World Knowledge Forgetting in Large Language Models via MoE-Style Plugin. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1932–1945. Bangkok, Thailand: Association for Computational Linguistics.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-Efficient Transfer Learning for NLP. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 2790–2799. PMLR.

Hu, E. J.; yelong shen; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Joshi, M.; Choi, E.; Weld, D.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Barzilay, R.; and Kan, M.-Y., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1601–1611. Vancouver, Canada: Association for Computational Linguistics.

Lee, K.; Chang, M.-W.; and Toutanova, K. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6086–6096. Florence, Italy: Association for Computational Linguistics.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4582–4597. Online: Association for Computational Linguistics.

Lin, J.; Tang, J.; Tang, H.; Yang, S.; Chen, W.-M.; Wang, W.-C.; Xiao, G.; Dang, X.; Gan, C.; and Han, S. 2024. AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration. In Gibbons, P.; Pekhimenko, G.; and Sa, C. D., eds., *Proceedings of Machine Learning and Systems*, volume 6, 87–100.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Meng, F.; Wang, Z.; and Zhang, M. 2024. PiSSA: Principal Singular Values and Singular Vectors Adaptation of Large Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Roy, O.; and Vetterli, M. 2007. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, 606–610.

Strang, G. 2022. *Introduction to linear algebra*. SIAM.

Tang, P.; Hu, X.; and Liu, Y. 2025. ADePT: Adaptive Decomposed Prompt Tuning for Parameter-Efficient Fine-tuning. In

The Thirteenth International Conference on Learning Representations.

Wang, H.; Li, Y.; Wang, S.; Chen, G.; and Chen, Y. 2025a. MiLoRA: Harnessing Minor Singular Components for Parameter-Efficient LLM Finetuning. In Chiruzzo, L.; Ritter, A.; and Wang, L., eds., *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 4823–4836. Albuquerque, New Mexico: Association for Computational Linguistics. ISBN 979-8-89176-189-6.

Wang, X.; Zheng, Y.; Wan, Z.; and Zhang, M. 2025b. SVD-LLM: Truncation-aware Singular Value Decomposition for Large Language Model Compression. In *The Thirteenth International Conference on Learning Representations*.

Wei, Y.; Wang, Z.; Liu, J.; Ding, Y.; and Zhang, L. 2024. Magicoder: Empowering Code Generation with OSS-Instruct. In Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; and Berkenkamp, F., eds., *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 52632–52657. PMLR.

Wu, C.; Gan, Y.; Ge, Y.; Lu, Z.; Wang, J.; Feng, Y.; Shan, Y.; and Luo, P. 2024. LLaMA Pro: Progressive LLaMA with Block Expansion. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6518–6537. Bangkok, Thailand: Association for Computational Linguistics.

Xu, C.; Sun, Q.; Zheng, K.; Geng, X.; Zhao, P.; Feng, J.; Tao, C.; Lin, Q.; and Jiang, D. 2024. WizardLM: Empowering Large Pre-Trained Language Models to Follow Complex Instructions. In *The Twelfth International Conference on Learning Representations*.

Yang, Y.; Li, X.; Zhou, Z.; Song, S. L.; Wu, J.; Nie, L.; and Ghanem, B. 2024. CorDA: Context-Oriented Decomposition Adaptation of Large Language Models for Task-Aware Parameter-Efficient Fine-tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yao, X.; Qian, H.; Hu, X.; Xu, G.; Liu, W.; Luan, J.; Wang, B.; and Liu, Y. 2025. Theoretical Insights into Fine-Tuning Attention Mechanism: Generalization and Optimization. In Kwok, J., ed., *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, 6830–6838. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Yu, L.; Jiang, W.; Shi, H.; YU, J.; Liu, Z.; Zhang, Y.; Kwok, J.; Li, Z.; Weller, A.; and Liu, W. 2024. MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.