Controlled Low-Rank Adaptation with Subspace Regularization for Continued Training on Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) exhibit remarkable capabilities in natural language processing but face catastrophic forgetting when learning new tasks, where adaptation to a new domain leads to a substantial decline in performance on previous tasks. In this paper, we propose Controlled LoRA (CLoRA), a subspace regularization method on LoRA structure. Aiming to reduce the scale of output change while introduce minimal constraint on model capacity, CLoRA imposes constraint on the direction of updating matrix's null space. Experimental results on commonly used LLM finetuning tasks reveal that CLoRA significantly outperforms existing LoRA subsequent methods on both in-domain and out-domain evaluations, highlighting the superority of CLoRA as a effective parameter-efficient finetuning method with catastrophic forgetting mitigating. Further investigation for model parameters indicates that CLoRA effectively balances the trade-off between model capacity and degree of forgetting.

1 Introduction

Large language models (LLMs) demonstrate remarkable capabilities in natural language tasks. However, when performing continued training on additional datasets, a key challenge may faced, known as catastrophic forgetting (McCloskey and Cohen, 1989), where adaptation to a new domain leads to a substantial decline in performance on previous tasks.

Existing approaches to mitigate catastrophic forgetting can be broadly categorized into data-based, architecture-based, and learning-based methods (Wang et al., 2023a). Data-based methods (de Masson D'Autume et al., 2019) primarily based on rehearsing prior training data, which raises data privacy concerns. Additionally, for LLMs, obtaining the necessary prior training data samples is challenging due to their training on massive datas. Architecture-based methods (Wang et al., 2023c; Razdaibiedina et al., 2023) introduce isolated parameters for each continued training stage for reducing interference. In contrast, learning-based methods train in the shared vector space, controlling learning process by adding regularization terms to the loss or employing specific optimization designs. Inference for architecure-based methods typically involves a selection process (Gurbuz and Dovrolis, 2022; Kang et al., 2022), which is more complex than that for learning-based methods. As continued trained LLMs are generally regarded as foundation models, flexibility is essential for their broader applications. Consequently, due to deployment considerations, learning-based methods are preferred over architecture-based methods for LLMs.

The core idea of learning-based methods is to constrain parameter updates, which aligns precisely with the Parameter-Efficient FineTuning (PEFT) research paradigm of LLMs. Although initially proposed for computational efficiency, PEFTs have demonstrated to learn less and forget less(Biderman et al., 2024), primarily due to their constrained model capacity. Notably, a well-established insight that related to learning-based methods in PEFT research is that LLMs are primarily finetuned within a specific low-rank subspace, this insight has led to the development of the Low-Rank Adaptation method (LoRA)(Hu et al., 2021).

However, LoRA imposes no restrictions on parameter updates beyond the low-rank constraint, and matrix perturbation theory suggests that even low-rank updates can significantly influence matrix properties (Sherman, 1949; Davis and Kahan, 1970). For instance, in an extreme case, it is theoretically possible to learn a model that eliminates all top-k principal components (optimal rank-k approximation) through a rank-k update, thus destroy most of the base model's ability. Therefore, LoRA would be benifit from more constraints for mitigat-

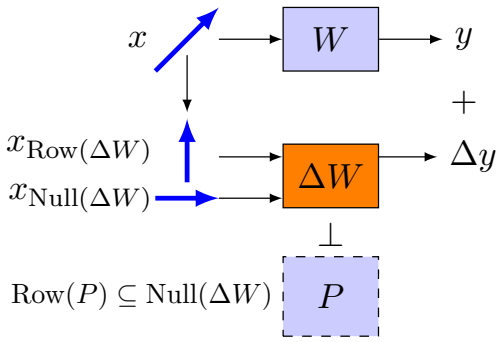

Figure 1: Illustration of the intuition behind our approach. For input x, the component in $\text{Null}(\Delta W)$ (null space of the updating matrix ΔW) would be ignored, the change of output Δy is obtained only from the component in $\text{Row}(\Delta W)$ (row space of ΔW, the orthogonal complement of $\text{Null}(\Delta W)$). CLoRA introduces a pre-defined subset of $\text{Null}(\Delta W)$ by imposing orthogonal regularization with pre-defined matrix P.

ing catastrophic forgetting. However, more constraints would reduce model capacity for updating, which influences the effectiveness of training. For instance, adding L2 regularization significantly restricts the norm of the updating matrix. Consequently, effective management of the capacity-forgetting balancing has become a major concern.

To address this concern, in this work, we propose Controlled LoRA (CLoRA), a subspace regularization method on LoRA structure. We start the design of CLoRA from the perspective of the null space of updating matrix. The intuition behind CLoRA is illustrated in Figure 1, where the output change Δy is derived from applying the updating matrix ΔW on the component of the input x that falls within the row space of ΔW, while components in the null space are ignored. Under this intuition, for reducing the scale of output change, options include reducing the scale of ΔW, and encouraging more input component fall in the null space of ΔW. The former is more related to model capacity, and for concerns of capacity-forgetting balancing, we focus on the latter.

The dimension of the null space for the updating matrix is directly determined by the rank of it, LoRA already addressed. A key factor remains, the direction of null space, which influence input components that fall in, but free-learned LoRA does not constraint. CLoRA constraint the direction of null space of updating matrix by introducing a pre-defined subset of that, this is implemented by orthogonal regularization with a pre-defined matrix. Unlike methods that impose restrictions on rank or norm, which significantly influence model capacity, CLoRA introduces constraint on the direction of the null space. Experimental results on commonly used LLM finetuning evaluations demonstrate that our proposed CLoRA outperforms existing methods on both in-domain downstream tasks and out-domain evaluations. This indicates the superiority of CLoRA as an effective approach for parameter-efficient finetuning with catastrophic forgetting mitigating. Additionally, we take analysis on parameters of the learned model, results show that CLoRA reduces the scale of output change with minimal impact on model capacity.

Our contributions are summarized as follows,

- We propose CLoRA, a subspace regularization method on LoRA structure, which serves as an advanced parameter-efficient finetuning technique with catastrophic forgetting mitigating for LLMs.
- Our proposed CLoRA demonstrates superior performance on commonly used LLM finetuning tasks compared to existing methods. It has also shown remarkable mitigating of catastrophic forgetting in out-domain evaluations.
- Parameter investigation results indicate that CLoRA effectively balances the trade-off between model capacity and degree of forgetting.

2 Related Works

2.1 Reducing Catastrophic Forgeting

Catastrophic forgetting is a significant challenge in various transfer learning scenarios, including continual learning (Wang et al., 2023a) and LLM continued training (Wu et al., 2024). In these settings, continued training on new tasks may impair abilities of the pre-trained model. Approaches for mitigating catastrophic forgetting can be broadly categorized into data-based, architecture-based and learning-based methods.

Data-based methods primarily based on rehearsal of prior training data or representation, (de Masson D'Autume et al., 2019) introduce an episodic memory for experience rehearsal, (Rebuffi et al., 2017; Chaudhry et al., 2019) selects previous training data for rehearsaling. For LLMs, acquiring the necessary prior training data is challenging due to the extensive amount of data used in their training. Instead, the concept of rehearsal is commonly adopted by mixing data from general domains for

LLM continued training. This approach is generally orthogonal to model-related methods, thus we will not discuss it further.

Architecture-based methods (Wang et al., 2023c; Razdaibiedina et al., 2023) introduce isolated parameters for each continued training stage to reduce interference. (Wang et al., 2023c) use isolated parameters for each task, and enables a selecting mechenism during inference. Progressive Prompts (Razdaibiedina et al., 2023) sequentially concatenates prompts for each task with previously learned prompts. These architecture-based methods generally require specific techniques for inference and continued training, resulting in a lack of flexibility, particularly in the context of LLMs.

Learning-based methods performs continued training in a shared vector space, controlling learning process by adding regularization term on loss or applying specific optimization designs. Orthogonal Gradient Descent (OGD) (Farajtabar et al., 2020) constrains the parameters to move within the orthogonal space defined by the gradients of previous tasks. Continual LoRA (C-LoRA)(Smith et al., 2024) and O-LoRA (Wang et al., 2023b) introduce regularization with previous continual learned parameters for reducing interference in the multi-stage training setting. Our proposed CLoRA imposes orthogonal regularization similar to O-LoRA, but the regularization matrix is not restricted to be the previous learned parameter, thus CLoRA can be used for one-stage continued training whereas O-LoRA cannot.

2.2 LoRA and Subspace Tuning

Parameter-Efficient FineTuning (PEFT) (Han et al., 2024) aims to tune models with minimizing computational resources, which is widely used for large-scale models including LLMs. Among these methods, LoRA (Hu et al., 2021) and its subsequent variants (Wang et al., 2024a; Liu et al., 2024) learn a low-rank decomposition for updating parameter matrices, and could be categorized into learning-based continued training method, which is the focus of our work.

The core insight of LoRA is to tune model within a low-rank subspace, and with no additional constraints imposed on this tuning subspace. Some subsequent works delve deeper into the tuning subspace to mitigate catastrophic forgetting for LLM continued training, MiLoRA (Wang et al., 2024a) and PiSSA (Meng et al., 2024) use singular value decomposition (SVD) components of the original parameters for LoRA initialization, with MiLoRA uses minor components while PiSSA uses major components; O-LoRA (Wang et al., 2023b) introduce orthogonal regularization for each LoRA subspace. Our proposed CLoRA also falls within this category, differing from the selection and utilization of the focused subspace.

Notation	Description
W	parameter matrix in base model
ΔW	updating of the parameter
x	input for W
y	output for W, $y = Wx$
Δy	output change, $\Delta y = \Delta W x$
$\|\|v\|\|$	L2 norm of vector v
$\|\|A\|\|$	L2 norm(largest singular value) of matrix A
r	rank of updating matrix
k	number of regularization vectors

Table 1: Notations.

3 Prelimilaries

3.1 Notations

The notations commonly used in this paper are summarized in table 1. We provide some additional notes here. While generally used for denote the input and output of the while model, we denote x, y as input and output to a single linear layer, represented by W. $||A||$ denotes L2 norm (largest singular value) in our paper, instead of Frobenius norm ($||A||_F = \sqrt{\sum A[i,j]^2}$). r and k are most important hyperparameters for CLoRA, r is the rank of updating matrix, which is used in all LoRA works, k is the number of regularization vectors(column of regularization matrix) in CLoRA.

3.2 Problem Definition

We aim to mitigate catastrophic forgetting in LLM continued training. Catastrophic forgetting menifest as performance decline on tasks from previous domain when training on new domain. To evaluate this, we conduct both in-domain tasks (demonstrating the effectiveness of training) and out-domain tasks (from previous domain, indicating the degree of forgetting). We conduct experiments in the one-stage tuning setting, specifically, we finetune a base LLM on one training dataset, and then take in-domain and out-domain evaluations. Note that there is no clear domain specific

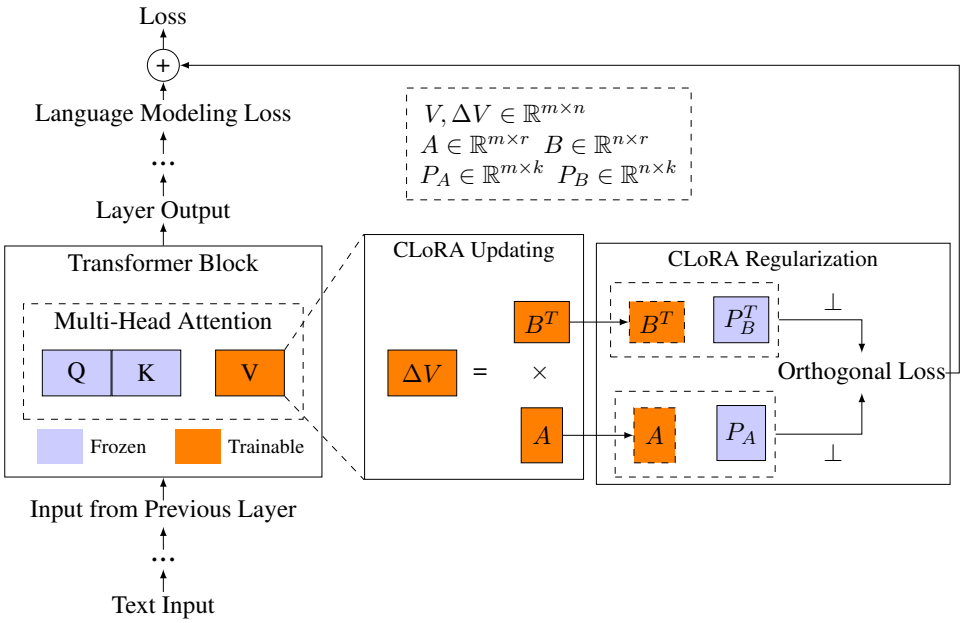

Figure 2: Illustration of CLoRA on typical decoder-only transformer based LLMs. LoRA updating is applied on v-proj in multi-head attention layer for each layer. CLoRA add orthogonal loss computes from trainable LoRA parameters (A and B) to the original language modeling loss.

for base LLMs, but benchmarks exist for evaluating the ability of LLMs on wide range of domains (Gao et al., 2024), and we take those with minimal overlap with training data for out-domain evaluation.

4 Method

In this section, we introduce Controlled Low-Rank Adaptation (CLoRA) method. We illustrate the application of CLoRA in transformer-based LLMs in Figure 2. CLoRA shares the same modeling structure with LoRA, but imposes on orthogonal regularization term computed using LoRA parameters into the loss function.

CLoRA Modeling Consistent with LoRA, CLoRA decomposes the updating for a parameter matrix W to a multiplication of two lor-rank matrices $\Delta W = AB^T$, where $W, \Delta W \in \mathbb{R}^{m\times n}$, $A \in \mathbb{R}^{m\times r}$, $B \in \mathbb{R}^{n\times r}$, $r \ll m, n$.

CLoRA computes orthogonal regularization for A and B^T with untrainable pre-defined matrix $P_A \in \mathbb{R}^{m\times k}$ and $P_B \in \mathbb{R}^{n\times k}$, where k is a hyperparameter controlling the size of regularization matrix, larger k introduces more constraint. The orthogonal regularization loss on one LoRA parameter A is defined as

$$L_{orth}(A, P_A) = \sum_{i,j} ||AP_A^T[i,j]||^2 \quad (1)$$

where $A \in \mathbb{R}^{m\times r}$, $P_A \in \mathbb{R}^{m\times k}$. $L_{orth}(A, P_A)$ regularize on orthogonality of every $(A[:,i], P_A[:,j])$ pairs. The final loss of CLoRA in a transformer-based LLM is defined as

$$L_{LM}(\Theta, input) + \lambda \sum_i (L_{orth}(A_i, P_{A_i}) + L_{orth}(B_i^T, P_{B_i}^T)) \quad (2)$$

where $L_{LM}(\Theta, input)$ is the original language model loss on text input and LLM parameters Θ, the summation on L_{orth} is over index of all trainable parameter matrices. λ controls the weighting of orthogonal loss, we set it to 1 as default.

Initialization Following LoRA(Hu et al., 2021), we initialize A with gaussian noise and B is zeros, ensuring ΔW is zero at the begining of training.

For the CLoRA regularization matrices, we suggest that they should have othogomal rows. Specifically, for regularization matrix $P \in \mathbb{R}^{m\times k}$, $||P[:,i]|| = 1$ for every i, and $P[:,i]P[:,j] = 0$ for $i \neq j$. This ensures uniform regularization. We propose several variants as follows,

- **Random Initialize:** Initialize with an orthogonal matrix(Saxe et al., 2014).
- **Initialize from SVD:** Previous works (Sharma et al., 2024; Wang et al., 2024a) have explored the roles of singular value decomposition (SVD)

components in LLM parameters. We adopt this intuition to initialize CLoRA regularization matrices from SVD. For a SVD decomposition of parameter $W = USV^T$ with rank r, where $W \in \mathbb{R}^{m\times n}$, $U \in \mathbb{R}^{m\times r}$, $S \in \mathbb{R}^{r\times r}$ is a diagonal matrix, $V \in \mathbb{R}^{n\times r}$. For CLoRA updating $\Delta W = AB^T$, we initialize the regularization matrix $P_A \in \mathbb{R}^{m\times k}$ as $U[:, \mathrm{s}]$, $P_B \in \mathbb{R}^{n\times k}$ as $V[:, \mathrm{s}]$, where s is a list of selecting index with length k.

5 Experiment and Analysis

In this section, we conduct experiments to evaluate and analyze our proposed CLoRA method. All evaluations are available in lm-eval (Gao et al., 2024) with MIT License. We aim to answer these research questions,

- **RQ1:** Does CLoRA perform effectively as a parameter-efficient finetuning method with catastrophic forgetting mitigating?
- **RQ2:** How do the size and selection of regularization matrices influence the performance of CLoRA?
- **RQ3:** How does CLoRA demonstrate superiority on capability-forgetting balancing?

5.1 Evaluating for Tuning on Commonsense Reasoning Dataset (RQ1)

To answer **RQ1**, we conduct experiments following the setting in the previous works on PEFT(Liu et al., 2024; Wang et al., 2024a). Specifically, we finetune LLaMA-2-7B (Touvron et al., 2023) with a dataset for commonsense reasoning. We evaluated the model on both in-domain downstream tasks to evaluate learning effectiveness and on out-domain tasks with minimal overlap with the training dataset to measure the degree of forgetting. We report partial results here, full results on each CLoRA variants and results on finetuning LLaMA-3-8B(AI@Meta, 2024) are reported in Appendix B.

5.1.1 Experiment Setup

Datasets and Tasks We use Commonsense170K (Hu et al., 2023) for finetuning. For in-domain evaluation, eight commonsense reasoning datasets are used, including BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag(HS) (Zellers et al., 2019), WinoGrande(WG) (Sakaguchi et al., 2019), ARC-e, ARC-c (Clark et al., 2018), and OBQA (Mihaylov et al., 2018). The tasks are formulated as multiple-choice problem, and we report accuracy based on the last checkpoint. For out-domain evaluations, BIG-Bench-Hard(BBH) (Suzgun et al., 2022) and MMLU-Pro (Wang et al., 2024b) are used. These benchmarks encompass challenging subsets of tasks across a wide range of domains and are widely employed for evaluating the capabilities of LLMs. Additionally, they include samples that are more complex than those in our training data, ensuring minimal overlap. We use lm-eval (Gao et al., 2024) for reporting out-domain evaluation.

Comparison Methods

- **LoRA** (Hu et al., 2021) is a widely-used parameter-efficient finetuning technique, and it serves as the foundation of our proposed CLoRA.
- **DoRA** (Liu et al., 2024) is a recent work on structure improvement of LoRA, we include it as a baseline for improved LoRA.
- **PiSSA** (Meng et al., 2024) and **MiLoRA** (Wang et al., 2024a) are two variants of LoRA, both employing SVD components for LoRA initialization, MiLoRA use minor components while PiSSA use major. Notably, MiLoRA can be categorized as a catastrophic forgetting mitigating method.
- **Reducing the updating rank(-r*):** Lower rank r imposes stricter constraints on the updating matrix. We maintain a consistent rank across all methods and consider variations in rank as a separate baseline.
- **L2 regularization(-L2)** introduces L2 regularization for trainable parameters, serving as a fundamental approach to limit updates.
- **CLoRA:** Random initialize variant of CLoRA, it is the basic setting of our method. We will include results for other variants in the subsequent section 5.2.

Hyperparameter We use the same hyperparameter configurations as (Wang et al., 2024a). Details are listed in Appendix A. Notably, we use 32 for updating matrix rank r as default for all methods. For CLoRA regularization matrix, we select k in $[128, 256, 512, 1024, 2048]$. For LoRA-L2, 1e-5 is used for weighting of L2 regularization, we note that 1e-4 is also tested, but too large for getting effective tuning.

Category	Method	BoolQ	PIQA	SIQA	HS	WG	ARC-e	ARC-c	OBQA	Avg.
ChatGPT	ChatGPT	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LoRAs	LoRA	71.9	80.9	78.9	90.3	83.5	83.0	70.2	80.8	79.9
	DoRA	73.0	81.9	80.3	90.2	82.8	84.6	69.4	81.8	80.5
	PiSSA	67.6	78.1	78.4	76.6	78.0	75.8	60.2	75.6	73.8
Reducing Forgetting	MiLoRA	71.5	82.0	80.0	91.0	83.0	82.3	68.9	81.2	80.0
	LoRA-r8	71.0	80.5	78.1	90.0	83.0	81.1	68.5	78.0	78.8
	LoRA-r16	71.0	81.8	78.9	90.3	81.1	83.1	69.7	82.2	79.8
	LoRA-L2	70.3	83.0	80.2	92.7	83.1	84.2	71.2	81.4	80.8
Ours	CLoRA	**73.7**	**84.5**	**80.9**	**94.5**	**85.9**	**88.1**	**75.9**	**86.0**	**83.7**

Table 2: In-domain results on commonsense reasoning evaluations, with accuracy scores (%) reported. **Bold** font indicates the highest performance for each dataset across the different PEFT methods.

5.1.2 Results and Analysis

In-domain Evaluation We report in-domain results in Table 2. LLaMA-2-7B cannot be directly applied to these tasks without finetuning, following previous works(Wang et al., 2024a), we include the ChatGPT baseline reported in (Liu et al., 2024) as a reference, which is obtained with gpt-3.5-turbo API using a zero-shot Chain of Thought (Wei et al., 2022). For CLoRA, we report the best result across all results for all k.

For in-domain evaluation, results demonstrate that CLoRA outperforms on all datasets, surpassing the best baseline by 2.9 points in average accuracy (%). These outcomes suggest that, although proposed primarily for mitigating catastrophic forgetting, CLoRA also performs as a effective PEFT method. We attribute this to the nature of LLM finetuning, as an instance of transfer learning, the performance of LLM finetuning is strongly correlated with the base model's ability, when catastrophic forgetting occurs during training, the base model's strength may diminish. Therefore, we hypothesize that a method with effective capacity-forgetting balancing would exhibit strong effectiveness in LLM finetuning.

Out-domain Evaluation We report out-domain results for CLoRA and baselines in Table 3. "base" denotes the result on the base model LLaMA-2-7B, "CLoRA" denotes the result of best model from in-domain evaluation.

In out-domain evaluation, all baselines underperform compared to the base model, highlighting the severe issue of catastrophic forgetting in this experimental setup. Notably, our proposed CLoRA not only outperforms all baselines, but also surpasses the base model's performance. We attribute this to the effective capacity-forgetting balancing for CLoRA, enabling the effective extraction of generally useful knowledge from the commonsense reasoning training dataset.

Method	BBH	MMLU-Pro
base	34.91	18.56
PiSSA	29.54	11.33
DoRA	28.24	11.67
LoRA	26.69	14.46
MiLoRA	25.14	17.74
LoRA-r8	26.90	14.58
LoRA-r16	26.73	11.54
LoRA-L2	32.93	16.59
CLoRA	**38.67**	**20.59**

Table 3: Out-domain results on LLM benchmarks, with accuracy scores (%) reported.

Answer for RQ1: The superior performance in both in-domain and out-domain evaluations demonstrates that our proposed CLoRA serves effectively as a parameter-efficient finetuning method with catastrophic forgetting mitigating.

5.2 Evaluating for different CLoRA variants and hyperparameters (RQ2)

To answer **RQ2**, we compare results for various CLoRA variants to demonstrate the impact of regularization matrix's size and selection.

Datasets and Tasks Same as in section 5.1, we finetune LLaMA-2-7B on Commonsense170K, and conduct in-domain evaluation on eight commonsense reasoning tasks and out-domain evaluation on two LLM benchmarks. We report average results for each evaluation. To ascertain whether the optimal k value is consistent across tasks, we additionally perform an experiment in the math domain.

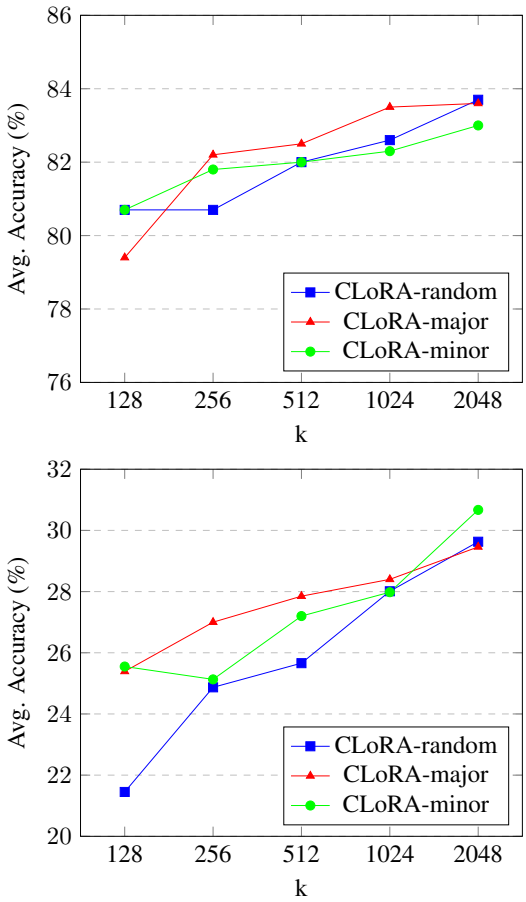

Figure 3: Performace for CLoRA on commonsense reasoning finetuning setting in-domain(above), out-domain(below) with different variant and k.

we finetune LLaMA-2-7B on the MetaMathQA (Yu et al., 2024), which contains 395K samples augmented from the training set of GSM8K (Cobbe et al., 2021) and MATH(Hendrycks et al., 2021). We use test set of GSM8K and MATH for evaluation and report the results on the last checkpoint.

Comparision Methods

- **Different k(-k*):** We test CLoRA with different regularization matrix size (k). Higher k imposes stricter constraints on the updating matrix.
- **CLoRA-random:** CLoRA with random initialization.
- **CLoRA-major:** Use SVD major components to initialize CLoRA regularization matrix.
- **CLoRA-minor:** Use SVD minor components to initialize CLoRA regularization matrix.

For math evaluation, PiSSA(Meng et al., 2024) and MiLoRA(Wang et al., 2024a) are included as baselines, we use CLoRA-major for CLoRA variant.

Method	GSM8K	MATH
LoRA	60.58	16.88
PiSSA	58.23	15.84
MiLoRA	63.53	17.76
CLoRA-k64	64.29	17.52
CLoRA-k128	**64.59**	**18.38**
CLoRA-k256	63.45	17.58

Table 4: Math evaluation on GSM8K and MATH, with accuracy scores (%) reported.

5.2.1 Results and Analysis

Commonsense Reasoning Setting We illustrate the performance for in-domain and out-domain evaluations with different k for commonsense reasoning setting as in Figure 3. The results indicate that for all CLoRA variants tested, as k increases, the scores of both in-domain and out-domain evaluation generally exhibit an upward trend. Despite similar performance across variants under the same k, CLoRA-major generally performs better, highlighting the advantage of preserving the well-learned major components (Wang et al., 2024a) of the base model.

Math Setting Results on math setting are reported in Table 4. CLoRA outperforms other baselines, and the optimal k value found to be around 128, differing from the 2048 in commonsense reasoning setting. This discrepancy can be attributed to the complexity of math tasks, which necessitate greater model capacity during finetuning. These findings suggest that adjusting k benefits the balancing of capacity and forgetting, demonstrating the flexibility of our approach.

Answer for RQ2: Results indicate that the choice of regularization matrix does influence the effectiveness of CLoRA, albeit not significantly. Generally, we recommend using random initialization (CLoRA-random) or initialization from major SVD components (CLoRA-major). Regarding the choice of k, as k increases, a trend of performance improvement followed by a decline is observed. Different tasks may correspond to different optimal k, we suggest that a smaller k could be used for more challenging tasks.

5.3 Understanding Capacity-Forgetting Balancing(RQ3)

To answer **RQ3**, we investigate the parameter of trained models to quantify the capacity-forgetting

balancing issue.

Measuring Model Capacity and Degree of Forgetting Consider that catastrophic forgetting primarily arises from output changes caused by parameter updating, the greater the impact of these updates, the more severe the catastrophic forgetting may be. We measure the degree of forgetting with the relative scale of output change in the parameter level, to be specific, for updating matrix ΔW, with input x, the relative scale of output change (denoted as F) is defined as

$$\mathrm{F}(\Delta W, x) = \frac{||\Delta W x||}{||x||} \tag{3}$$

We highlight the role of x in the measurement of "F", as it reflects the real world use case. Specifically, we sample 100 text data from test set and collect input x for each parameter from the model forward pass.

To measure model capacity, we note that there is a gap between theoretical capacity of a model (Abu-Mostafa, 1989) and the practical outcome of the learned model. Therefore, we delegate the measurement of model capacity to the scale of the parameters in the learned model. Specifically, we measure the L2 norm $||\Delta W||$ for each updating parameter matrix.

Results and Analysis We report the measurements averaged over all tokens and all updating parameters in Table 5. All models use LoRA rank r of 32 unless specified otherwise.

The "reference" row is computed using the LoRA trained model, noting the output scale of original parameter W instead of ΔW. Compared with "reference" and "LoRA", the difference of "F" is not very far, suggesting that LoRA training indeed introduces significant output change, thus still prone to catastrophic forgetting.

For "MiLoRA", although intuitively promising, without effective control during training, it did not mitigate catastrophic forgetting, as evidenced by both downstream evaluations (in section 5.1) and the similar "F" and ΔW with LoRA.

For LoRA with lower rank (r8 and r16), after training, with $||\Delta W||$ indicates the reduction of capacity, the scale of output change does not show a decrease. Although theoretically, reducing the rank of the update matrix can increase the dimension of the null space and help to reduce the scale of output change, results not show this case. This suggests that altering r may not a effctive way to alter forgetting.

Method	$\|\|\Delta W\|\|$	F
reference		2.42
LoRA	22.63	0.79
MiLoRA	24.32	0.92
LoRA-r16	12.70	1.03
LoRA-r8	6.45	0.95
LoRA-L2	2.07	0.29
CLoRA-k128	10.84	0.36
CLoRA-k256	10.25	0.34
CLoRA-k512	8.19	0.27
CLoRA-k1024	6.64	0.21
CLoRA-k2048	5.00	0.14

Table 5: Measuing of model updating capacity($||\Delta W||$, larger for more capacity) and degree of forgetting(F, lower for less forgetting) for trained models.

For LoRA-L2, "F" indicates that it indeed mitigate forgetting, but in a large cost of capacity, demonstrated by the very small $||\Delta W||$.

For our proposed CLoRA, "F" shows a significantly reduce the scale of output change, while a relatively larger $||\Delta W||$ is maintained. This indicates that CLoRA minimizes catastrophic forgetting caused by large updates while having a subtle impact on model capacity. Thus we answer **RQ3** that CLoRA performs effectively on capacity-forgetting balancing.

6 Conclusion

In this paper, we introduce Controlled Low-Rank Adaptation(CLoRA), a simple yet effective parameter-efficient finetuning method for LLMs that mitigates catastrophic forgetting. We investigate the effectiveness of CLoRA on commonsense reasoning and math reasoning tasks. Experiment results demonstrate that CLoRA consistently outperforms previous LoRA-based methods on in-domain evaluations, highlighting the effectiveness of CLoRA as a parameter-efficient finetuning method. Additionally, out-domain evaluations demonstrate that CLoRA exhibits strong capabilities in mitigating catastrophic forgetting. Further investigation for model parameters indicates that CLoRA effectively balances the trade-off between model capacity and degree of forgetting.

7 Limitations

There are still several limitations that we reserve for future work: 1) While we have noted that our proposed CLoRA would benefit from the dedicated selection of regularization matrix to retain specific model abilities, we did not go further into this aspect. In this paper, we only explored the SVD initialization variant alongside the basic random initialization. 2) We delegate the measurement of model capacity and degree of forgetting to simple measurement of scale. Although these measurements reveal significant differences between CLoRA and previous works, we believe that further investigation would aid in the design of methods with stronger capacity-forgetting balancing capability.

References

Yaser S Abu-Mostafa. 1989. The vapnik-chervonenkis dimension: Information versus complexity in learning. *Neural Computation*, 1(3):312–317.

AI@Meta. 2024. Llama 3 model card.

Dan Biderman, Jose Gonzalez Ortiz, Jacob Portes, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. 2024. Lora learns less and forgets less. *Preprint*, arXiv:2405.09673.

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

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. 2019. On tiny episodic memories in continual learning. *Preprint*, arXiv:1902.10486.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

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. *arXiv:1803.05457v1*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Chandler Davis and W. M. Kahan. 1970. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46.

Cyprien de Masson D'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems*, 32.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. 2020. Orthogonal gradient descent for continual learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3762–3773. PMLR.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Mustafa B Gurbuz and Constantine Dovrolis. 2022. NISPA: Neuro-inspired stability-plasticity adaptation for continual learning in sparse networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8157–8174. PMLR.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

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

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy 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*, pages 5254–5276, Singapore. Association for Computational Linguistics.

Haeyong Kang, Rusty John Lloyd Mina, Sultan Rizky Hikmawan Madjid, Jaehong Yoon, Mark Hasegawa-Johnson, Sung Ju Hwang, and Chang D. Yoo. 2022. Forget-free continual learning with winning subnetworks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 10734–10750. PMLR.

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. *Preprint*, arXiv:2402.09353.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. 2024. Pissa: Principal singular values and singular vectors adaptation of large language models. *Preprint*, arXiv:2404.02948.

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*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. In *The Eleventh International Conference on Learning Representations*.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. 2014. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *Preprint*, arXiv:1312.6120.

Pratyusha Sharma, Jordan T. Ash, and Dipendra Misra. 2024. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. In *The Twelfth International Conference on Learning Representations*.

J. Sherman. 1949. Adjustment of an inverse matrix corresponding to changes in the elements of a given column or row of the original matrix.

James Seale Smith, Yen-Chang Hsu, Lingyu Zhang, Ting Hua, Zsolt Kira, Yilin Shen, and Hongxia Jin. 2024. Continual diffusion: Continual customization of text-to-image diffusion with c-loRA. *Transactions on Machine Learning Research*.

Mirac Suzgun, Nathan Scales, Nathanael Sch?rli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *Preprint*, arXiv:2210.09261.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Hanqing Wang, Zeguan Xiao, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. 2024a. Milora: Harnessing minor singular components for parameter-efficient llm finetuning. *Preprint*, arXiv:2406.09044.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2023a. A comprehensive survey of continual learning: Theory, method and application. *Preprint*, arXiv:2302.00487.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023b. Orthogonal subspace learning for language model continual learning. *Preprint*, arXiv:2310.14152.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark (published at neurips 2024 track datasets and benchmarks). *Preprint*, arXiv:2406.01574.

Zhicheng Wang, Yufang Liu, Tao Ji, Xiaoling Wang, Yuanbin Wu, Congcong Jiang, Ye Chao, Zhencong Han, Ling Wang, Xu Shao, and Wenqiu Zeng. 2023c. Rehearsal-free continual language learning via efficient parameter isolation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10933–10946, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. *Preprint*, arXiv:2402.01364.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*.

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 Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

A Detailed Experiment Setups

A.1 Hyperparameter Settings

Table 6 shows our detailed hyperparameters. This setting follows MiLoRA(Wang et al., 2024a) and DoRA(Liu et al., 2024).

A.2 Computation Environment

All of our experiments are conducted on 8 NVIDIA A800 GPUs. All methods for LoRA subsequents use Huggingface peft library[1], training is conducted using trainer in Huggingface transformers library[2], with DeepSpeed ZeRO(Rajbhandari et al., 2020) intergration.

B Full Experiment Results

B.1 Full Results on Commonsense Finetuning

We report the full results that we conducted on in-domain evaluation(Table 7) and out-domain evaluation(Table 8) for commonsense reasoning finetuning. All models use LoRA rank r of 32 unless specified otherwise.

[1] https://github.com/huggingface/peft
[2] https://github.com/huggingface/transformers

Hyperparameter	CS	Math
LoRA rank r	32	64
LoRA α	64	128
Dropout	0.05	
Optimizer	AdamW	
LR for LLaMA-2-7B	3e-4	
LR for LLaMA-3-8B	1e-4	
LR Scheduler	Linear	
Batch Size	16	
Warmup Steps	100	
Epochs	3	
LoRA target modules	query, key, value, MLP up, MLP down	

Table 6: Hyperparameters for commonsense reasooning (CS) and Math settings.

Model	PEFT	BoolQ	PIQA	SIQA	HS	WG	ARC-e	ARC-c	OBQA	Avg.
ChatGPT	-	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LLaMA-2-7B	LoRA	71.9	80.9	78.9	90.3	83.5	83.0	70.2	80.8	79.9
	PiSSA	67.6	78.1	78.4	76.6	78.0	75.8	60.2	75.6	73.8
	MiLoRA	71.5	82.0	80.0	91.0	83.0	82.3	68.9	81.2	80.0
	DoRA	73.0	81.9	80.3	90.2	82.8	84.6	69.4	81.8	80.5
	LoRA-r8	71.0	80.5	78.1	90.0	83.0	81.1	68.5	78.0	78.8
	LoRA-r16	71.0	81.8	78.9	90.3	81.1	83.1	69.7	82.2	79.8
	LoRA-L2-0.0001	-	-	-	-	-	-	-	-	-
	LoRA-L2-0.00001	70.3	83.0	80.2	92.7	83.1	84.2	71.2	81.4	80.8
	CLoRA-random-k128	72.7	84.1	77.7	91.6	83.0	85.3	69.9	81.6	80.7
	CLoRA-random-k256	71.3	83.2	79.1	92.4	83.2	84.5	71.0	81.0	80.7
	CLoRA-random-k512	72.8	83.0	79.5	93.0	83.9	85.7	73.0	84.8	82.0
	CLoRA-random-k1024	73.3	84.8	79.6	91.1	**86.1**	86.9	73.1	85.6	82.6
	CLoRA-random-k2048	73.7	84.5	**80.9**	94.5	85.9	**88.1**	75.9	**86.0**	**83.7**
	CLoRA-major-k128	72.4	81.9	77.9	83.9	82.4	84.4	70.0	82.6	79.4
	CLoRA-major-k256	73.2	83.5	79.6	93.0	83.3	88.1	72.6	84.2	82.2
	CLoRA-major-k512	73.6	83.7	79.9	93.4	83.9	86.4	73.0	86.0	82.5
	CLoRA-major-k1024	73.2	**85.5**	80.5	94.3	85.7	87.2	75.9	85.4	83.5
	CLoRA-major-k2048	**73.9**	84.8	80.6	**95.0**	85.3	87.7	**76.5**	84.6	83.6
	CLoRA-minor-k128	71.5	82.7	78.7	91.8	83.2	85.0	70.9	81.6	80.7
	CLoRA-minor-k256	72.6	83.5	80.2	91.3	85.4	85.4	72.1	83.6	81.8
	CLoRA-minor-k512	73.0	84.0	80.1	93.1	82.0	86.4	72.9	84.4	82.0
	CLoRA-minor-k1024	73.1	83.7	79.2	93.7	84.8	87.1	73.2	83.2	82.3
	CLoRA-minor-k2048	72.9	84.2	80.8	93.7	85.3	87.2	73.5	**86.0**	83.0
LLaMA-3-8B	LoRA	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
	PiSSA	67.1	81.1	77.2	83.6	78.9	77.7	63.2	74.6	75.4
	MiLoRA	68.8	86.7	77.2	92.9	85.6	86.8	75.5	81.8	81.9
	DoRA	74.6	89.3	79.9	95.5	85.6	90.5	80.4	85.8	85.2
	CLoRA-random-k128	75.5	89.1	81.6	95.9	87.9	92.6	81.8	86.8	86.4
	CLoRA-random-k256	75.3	88.8	81.4	85.7	88.7	92.7	82.3	88.4	85.4
	CLoRA-random-k512	75.9	89.3	82.6	96.3	**88.9**	92.1	82.9	86.8	86.9
	CLoRA-random-k1024	**76.5**	89.1	82.1	96.3	88.6	93.0	81.7	**90.0**	87.2
	CLoRA-random-k2048	76.2	**90.0**	**82.7**	**96.6**	88.8	**93.3**	**83.4**	89.2	**87.5**

Table 7: In-domain results on commonsense reasoning evaluations, with accuracy scores (%) reported. **Bold** font indicates the highest performance for each dataset across the different PEFT methods for each base model.

Model	PEFT	BBH	MMLU-Pro	Avg.
LLaMA-2-7B	-	34.91	18.56	26.74
	LoRA	26.69	14.46	20.58
	PiSSA	29.54	11.33	20.44
	MiLoRA	25.14	17.74	21.44
	DoRA	28.24	11.67	19.96
	LoRA-r8	26.90	14.58	20.74
	LoRA-r16	26.73	11.54	19.13
	LoRA-L2-0.00001	32.93	16.59	24.76
	CLoRA-random-k128	30.82	12.07	21.45
	CLoRA-random-k256	31.92	17.81	24.87
	CLoRA-random-k512	34.32	17.00	25.66
	CLoRA-random-k1024	36.49	19.52	28.01
	CLoRA-random-k2048	38.67	**20.59**	29.63
	CLoRA-major-k128	32.69	18.09	25.39
	CLoRA-major-k256	35.11	18.89	27.00
	CLoRA-major-k512	35.81	19.88	27.85
	CLoRA-major-k1024	37.06	19.73	28.40
	CLoRA-major-k2048	38.83	20.08	29.46
	CLoRA-minor-k128	34.06	17.03	25.55
	CLoRA-minor-k256	33.16	17.11	25.13
	CLoRA-minor-k512	35.42	18.97	27.20
	CLoRA-minor-k1024	37.08	18.87	27.98
	CLoRA-minor-k2048	**40.96**	20.37	**30.67**

Table 8: Out-domain results on two LLM benchmarks, with accuracy scores (%) reported. **Bold** font indicates the highest performance for each benchmark across all methods.