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

Yuheng Lu[1], Bingshuo Qian[1], Caixia Yuan[1], Huixing Jiang[2], Xiaojie Wang[1,*]
[1]School of Artificial Intelligence, Beijing University of Posts and Telecommunications
[2]LI Auto Inc.
{yuheng.lu, bsqian, yuancx, xjwang}@bupt.edu.cn, jianghuixing@lixiang.com

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 introducing minimal constraint on model capacity, CLoRA imposes constraints on the direction of updating matrix's null space. Experimental results on one-stage LLM finetuning tasks and continual learning settings highlight the superiority of CLoRA as an 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. The code for implementing CLoRA will be publicly available[1].

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 be 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) are 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 data. 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 the learning process by adding regularization terms to the loss or employing specific optimization designs. Inference for architecture-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 Fine-Tuning (PEFT) research paradigm of LLMs. Although initially proposed for computational efficiency, PEFTs have been demonstrated to learn less and forget less(Biderman et al., 2024), primarily due to their constrained model capacity. Notably, a well-established insight 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 destroying most of the base model's ability. Therefore, LoRA

*Corresponding author
[1]https://github.com/sutakori/CLoRA

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 19165–19181

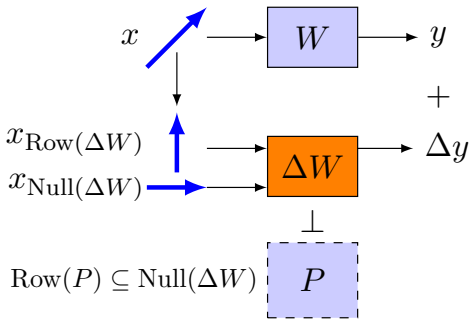

Figure 1: Illustration of the intuition behind our approach. For input x, the component in $\mathrm{Null}(\Delta W)$ (null space of the updating matrix ΔW) would be ignored, the change of output Δy is obtained only from the component in $\mathrm{Row}(\Delta W)$ (row space of ΔW, the orthogonal complement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

would benefit from more constraints for mitigating 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, which 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 subspace, 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. We take experiments on commonly used one-stage LLM finetuning evaluations and continual learning evaluations, results indicate 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, which serves as an advanced parameter-efficient finetuning technique with catastrophic forgetting mitigating for LLMs.
- Our proposed CLoRA demonstrates superior performance on both in-domain and out-domain evaluation in commonly used one-stage LLM finetuning setting. Additionally, it shows remarkable mitigating of catastrophic forgetting in continual learning setting.
- Parameter investigation results indicate that CLoRA effectively balances the trade-off between model capacity and degree of forgetting.

2 Related Works

2.1 Mitigating Catastrophic Forgetting

Catastrophic forgetting is a significant challenge in various transfer learning scenarios, including continual learning (Wang et al., 2023a) and LLM finetuning (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 rehearsing. 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 gener-

ally 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 mechanism 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 the learning process by adding a regularization term on loss or applying specific optimization designs. Notably, O-LoRA (Wang et al., 2023b) introduce regularization with previous continually 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 previously learned parameter, thus CLoRA can be used for one-stage continued training whereas O-LoRA not.

2.2 LoRA and Subspace Tuning

Parameter-Efficient Fine-Tuning (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
$\|\|A\|\|_F$	Frobenius norm of matrix A
r	rank of updating matrix
k	number of regularization vectors

Table 1: Notations.

3 Preliminaries

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 whole 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

Catastrophic forgetting manifest as performance decline on tasks from previous domain when training on new domain. In this work, we aim to mitigate catastrophic forgetting in LLM finetuning and continual learning settings.

3.2.1 LLM Finetuning

In this setting, we conduct experiments on one-stage LLM finetuning, 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) for LLM finetuning. Specifically, we finetune a base LLM on one training dataset, then take in-domain and out-domain evaluations. Note that there is no clear domain specific for base LLMs, but bench-

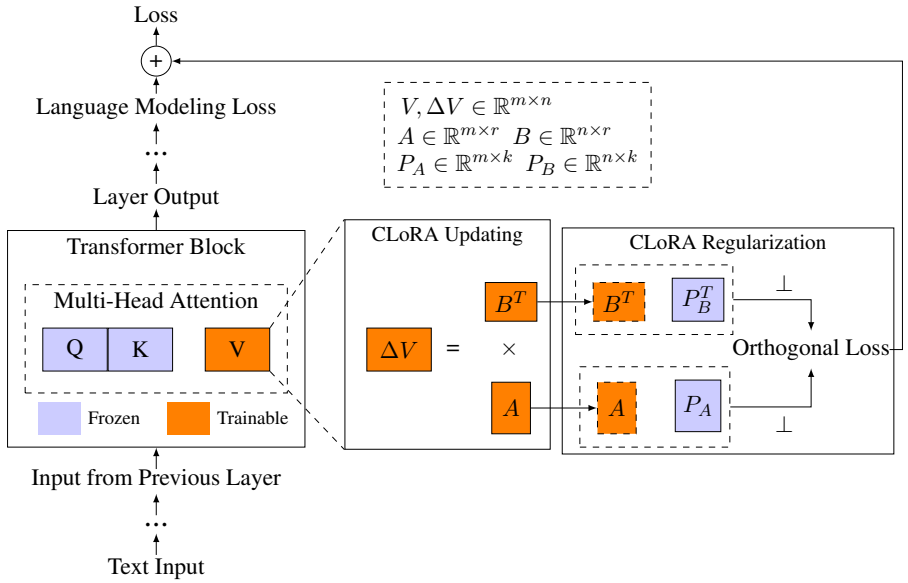

Figure 2: Illustration of CLoRA on typical decoder-only transformer based LLMs. LoRA updating is applied on v-proj in multi-head attention layer for each layer. CLoRA add orthogonal loss computes from trainable LoRA parameters (A and B) to the original language modeling loss.

marks 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.

3.2.2 Continual Learning

Continual learning focuses on developing learning algorithms to accumulate knowledge on non-stationary data(Wang et al., 2023b). In this setting, we conduct experiments for multi-stage finetuning. Specifically, we finetune the model on a sequence of tasks $D_1, \ldots, D_t$, where each task D_t contains a pair of train and test datasets $D_t = (D_t^{train}, D_t^{test})$. The t-th model with finetuned sequentially on $D_1^{train}, \ldots, D_{t-1}^{train}$ is tested over all previous test datasets $D_1^{test}, \ldots, D_{t-1}^{test}$.

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 low-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 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. P_A and P_B are untrainable, and fixed during training. The orthogonal regularization loss on one LoRA parameter A is defined as

$$L_{orth}(A, P_A) = ||A^T P_A||_F^2 \quad (1)$$

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 $L_{LM}(\Theta, x)$ is the original language model loss on text input x with 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 with zeros, ensuring ΔW is zero at the beginning of training.

For the CLoRA regularization matrices, following the principle of Occam's Razor, we adopt the

simplest random initialization here. For uniform regularization over each row in regularization matrices, we suggest using orthogonal initialization. Specifically, for regularization matrix $P \in \mathbb{R}^{m \times k}$, $||P[:, i]|| = 1$ for every i, and $P[:, i]P[:, j] = 0$ for $i \neq j$.

Training efficiency The vanilla transformer implementation exhibits time complexities of $O(n^2d + nd^2)$ for the attention layer and $O(nd^2)$ for the FFN layer, where n represents sequence length and d denotes hidden dimension size. As the base for CLoRA, LoRA performs as an efficient training method by reducing the gradient computation for origin parameter Ws, while introduces computation overhead with time complexity $O(ndr)$, which is minor as $r \ll d, n$ typically holds.

For our proposed CLoRA, additional computing overhead is paid for regularization computation, with time complexity of $O(kdr)$, which is minor compared to LoRA overhead when $k \ll n$. And even with large k proportional to d, $O(d^2r)$ for CLoRA additional overhead is minor compared to the transformer as $r \ll n$ generally. Thus, we claim that CLoRA preserves the computational advantages of LoRA and introduces minor additional overheads.

In terms of memory efficiency, CLoRA's frozen regularization matrix inherits LoRA's key advantage, specifically, it avoids storing full-sized parameters in optimizer states. This makes CLoRA as a memory-efficient training method.

5 Experiments and Analysis

5.1 One-Stage LLM Finetuning

In this section, we conduct experiments on one-stage LLM finetuning to evaluate our proposed CLoRA as a parameter-efficient finetuning method. We aim to answer the following research questions,

- **RQ1:** Does CLoRA perform effectively as a parameter-efficient finetuning method for LLMs with catastrophic forgetting mitigating?
- **RQ2:** How the size of regularization matrix k influence the performance of CLoRA? Does it differ across tasks?
- **RQ3:** How does CLoRA demonstrate superiority on capability-forgetting balancing?

5.1.1 Datasets and Tasks

Following previous works on PEFT(Liu et al., 2024; Wang et al., 2024a), we conduct experiments on commonsense reasoning tasks and math tasks.

Commonsense Reasoning Setting 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 (Zellers et al., 2019), WinoGrande (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 (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), available with MIT License, for reporting out-domain evaluation.

Math Setting We use MetaMathQA (Yu et al., 2024) for finetuning, 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.

5.1.2 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

Method	In-domain									Out-domain	
	BQ	PQ	SQ	HS	WG	ACe	ACc	OQ	Avg.	BBH	MMLU
LLaMA2-7b	-	-	-	-	-	-	-	-	-	34.91	18.56
LoRAs											
LoRA	71.9	80.9	78.9	90.3	83.5	83.0	70.2	80.8	79.9	26.69	14.46
DoRA	73.0	81.9	80.3	90.2	82.8	84.6	69.4	81.8	80.5	28.24	11.67
PiSSA	67.6	78.1	78.4	76.6	78.0	75.8	60.2	75.6	73.8	29.54	11.33
Reducing Forgetting											
MiLoRA	71.5	82.0	80.0	91.0	83.0	82.3	68.9	81.2	80.0	25.14	17.74
LoRA-r8	71.0	80.5	78.1	90.0	83.0	81.1	68.5	78.0	78.8	26.90	14.58
LoRA-r16	71.0	81.8	78.9	90.3	81.1	83.1	69.7	82.2	79.8	26.73	11.54
LoRA-L2	70.3	83.0	80.2	92.7	83.1	84.2	71.2	81.4	80.8	32.93	16.59
Ours											
CLoRA-k128	72.7	84.1	77.7	91.6	83.0	85.3	69.9	81.6	80.7	30.82	12.07
CLoRA-k256	71.3	83.2	79.1	92.4	83.2	84.5	71.0	81.0	80.7	31.92	17.81
CLoRA-k512	72.8	83.0	79.5	93.0	83.9	85.7	73.0	84.8	82.0	34.32	17.00
CLoRA-k1024	73.3	**84.8**	79.6	91.1	**86.1**	86.9	73.1	85.6	82.6	36.49	19.52
CLoRA-k2048	**73.7**	84.5	**80.9**	**94.5**	85.9	**88.1**	**75.9**	**86.0**	**83.7**	**38.67**	**20.59**

Table 2: Results for our proposed CLoRA and baselines for in-domain commonsense reasoning evaluations and out-domain LLM benchmarks, with accuracy scores (%) reported. **Bold** font indicates the highest performance for each task across all compared PEFT methods.

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:** Our proposed CLoRA method, with random initialized regularization matrix.

5.1.3 Experimental Configuration

We use the same base LLM choice LLaMA-2-7B (Touvron et al., 2023) and hyperparameter configurations as (Wang et al., 2024a). Details are listed in Appendix A. Notably, we use 32 (commonsense reasoning) and 64 (math) for updating matrix rank r as default for all methods if not explicitly specified. For the size of CLoRA regularization matrix, we select k in $[128, 256, 512, 1024, 2048]$ for commonsense reasoning and $[64, 128, 256]$ for more challenging math setting. 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 finetuning. We report results finetuned on LLaMA-2-7B here, more results are listed in Appendix A.

5.1.4 Main Results (RQ1)

For commonsense reasoning setting, we report the results of in-domain evaluation and out-domain LLM benchmarks in Table 2. The results demonstrate that CLoRA outperforms on all datasets, surpassing the best baseline for in-domain evaluation by an average accuracy of 2.9 points. Results for math setting also demonstrate the superiority of CLoRA over previous LoRA baselines (Table 3).

These outcomes suggest that, although primarily proposed for mitigating catastrophic forgetting, CLoRA also serves as an effective PEFT method. We attribute this to the nature of LLM finetuning, which is 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 claim that a method with effective capacity-forgetting balancing would exhibit strong effectiveness in LLM finetuning.

For out-domain evaluation, results show that all baselines underperform the base model, highlighting the severe issue of catastrophic forgetting in this setup. Notably, our proposed CLoRA not only outperforms all baselines by a significant margin but also surpasses the base model's performance. We attribute this to CLoRA's effective capacity-forgetting balancing, which enables the extraction of generally useful knowledge from the commonsense reasoning training dataset.

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

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. Thus, we answer **RQ1**.

5.1.5 Influences of CLoRA Hyperparameters (RQ2)

The size of the regularization matrix k is a crucial hyperparameter in CLoRA, balancing the trade-off between model capacity and the degree of forgetting. We focus here on how k influence the performance of finetuning LLM with CLoRA, and investigate whether the optimal k is consistent across tasks.

In commonsense reasoning setting, results show that larger k leads to better performance in both in-domain and out-domain evaluations (Table 2). In math setting, unlike the upward trend in commonsense reasoning setting, performance decreases when k exceeds 128(Table 3). We attribute this discrepancy to the complexity of math tasks, which require greater model capacity during finetuning.

Empirical results support the intuitive claim that larger k imposes more restrictions on updates, which helps to mitigate catastrophic forgetting but potentially limiting finetuning model capacity and harming performance.

Thus, we answer **RQ2** by demonstrating that the optimal k depends on task complexity. Notably, our proposed CLoRA provides flexibility in balancing capacity and forgetting by adjusting k, we suggest choosing a smaller k for more challenging tasks.

For the scale of regularization loss λ, we observe that the constraint effectiveness of CLoRA is not sensitive to that. CLoRA generally effectively enforces the orthogonal constraint, achieving near-complete suppression of updates along the constrained directions, and under the default $\lambda = 1$ configuration, satisfactory task performances are generally enabled.

Method	$\|\|\Delta W\|\|$	$\mathbb{F}_\Delta$	$\mathbb{F} \uparrow$
reference		2.42	34.91
LoRA	22.63	0.79	26.69
MiLoRA	24.32	0.92	25.14
LoRA-r16	12.70	1.03	26.73
LoRA-r8	6.45	0.95	26.90
LoRA-L2	2.07	0.29	32.93
CLoRA-k128	10.84	0.36	30.82
CLoRA-k256	10.25	0.34	31.92
CLoRA-k512	8.19	0.27	34.32
CLoRA-k1024	6.64	0.21	36.49
CLoRA-k2048	5.00	0.14	38.67

Table 4: Measuring model updating capacity($||\Delta W||$, larger indicates more capacity) and degree of forgetting for trained models. Measuring for degree of forgetting includes both mechanism oriented ($\mathbb{F}_\Delta$, lower indicates less forgetting) and performance oriented ($\mathbb{F}$, higher for less forgetting).

5.1.6 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 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, higher $||\Delta W||$ indicates potential more capacity. We use L2 norm (largest singular value) for highlighting the theoretical scale of output change, $||\Delta y|| = ||\Delta W x|| \leq ||\Delta W|| \cdot ||x||$, which reflect the theoretical effectiveness of output updates, and we delegate model capacity with it.

Measuring Degree of Forgetting (Performance Oriented) We use the out-domain evaluation on BBH for measuring degree of forgetting for each model, denoted as $\mathbb{F}$, which is aligned with the results in Table 2. $\mathbb{F}$'s performance-oriented measurement directly corresponds to the theoretical definition of catastrophic forgetting.

Measuring Degree of Forgetting (Mechanism Oriented) Beside the performance oriented measurement of forgetting $\mathbb{F}$, we try to find a mechanism oriented measurement so that reflect how the

CLoRA benefits.

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 would be. We thus 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 $\mathbb{F}$) is defined as

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

We sample 100 real world data and performs model forward pass for getting x in the measurement of $\mathbb{F}_{\Delta}(\Delta W, x)$, and the final $\mathbb{F}_{\Delta}$ is the average along each ΔW and x.

Results and Analysis We report the measurements averaged over all tokens and all updating parameters in Table 4. All models use LoRA rank r of 32 unless specified otherwise.

The **reference** row is computed using the LoRA trained model, we note that the $||\mathbb{F}_{\Delta}||$ denotes output scale of original parameter W instead of ΔW. Compared with the reference, LoRA's $\mathbb{F}_{\Delta}$ is large, suggesting that LoRA training indeed introduces significant output change, thus prone to catastrophic forgetting. Drop on $\mathbb{F}$ also indicates the forgetting.

For **MiLoRA**, although intuitively promising, without effective control during training, it did not mitigate catastrophic forgetting, as evidenced by $\mathbb{F}$, also by the similar $\mathbb{F}_{\Delta}$ and ΔW with LoRA.

For **LoRA with lower rank** (r8/16), after training, with $||\Delta W||$ indicates the reduction of capacity, $\mathbb{F}_{\Delta}$ and $\mathbb{F}$ does not show superiority. 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 an effective way to alter forgetting.

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

For our proposed **CLoRA**, $\mathbb{F}$ demonstrates a superior catastrophic forgetting mitigating. And $\mathbb{F}_{\Delta}$ shows a significant reduction in 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.

5.2 Continual Learning

5.2.1 Experimental Setup

To demonstrate the effectiveness of CLoRA for continual learning(CL) setting, we conduct experiments on standard CL benchmark and more challenging large number of tasks benchmark, following the experiment setup of O-LoRA(Wang et al., 2023b).

Datasets and Tasks The standard CL benchmark consists of five text classification datasets(Zhang et al., 2015). The large number of tasks benchmark consists of 15 datasets (Razdaibiedina et al., 2023), include tasks for natural language understanding and text classification. Task samples follow previous work (Wang et al., 2023b). Details for tasks are listed in Appendix B.

Comparison Methods We compare CLoRA with normal finetuning baselines and previous CL methods. We include non CL results that train separate model for each task (**PerTaskFT**) and multitask learning (**MTL**) as reference.

- **Normal Finetuing** baselines include sequentially training on same parameter space with full parameter finetune (**SeqFT**) and LoRA (**SeqLoRA**), and incremental learning of new LoRA parameters on a sequential series of tasks.

- **Continual Learning** methods include data-based methods **Replay**; architecture-based methods **L2P**(Wang et al., 2022), **LFPT5**(Qin and Joty, 2022)**O-LoRA**(Wang et al., 2023b); and learning-based methods **EWC**(Kirkpatrick et al., 2017), **LwF**(Leibe et al., 2016). We include two recent methods following O-LoRA, **LC-BL**(Qiao and Mahdavi) and **AM-LoRA**(Liu et al.), with improvment for inter task modeling. Details for these methods are listed in Appendix B.

- **Proposed CLoRA** includes setting that combine the design of O-LoRA that further performs regularization for previous learned LoRAs (+ O-LoRA).

Experimental Configuration Following O-LoRA(Wang et al., 2023b), we use T5-large as base model, and finetune on each task with specified order(Appendix B). We train each task

	Standard CL Benchmark				Large Number of Tasks			
Method	Order-1	Order-2	Order-3	avg.	Order-4	Order-5	Order-6	avg.
LoRAs								
SeqFT	18.9	24.9	41.7	28.5	7.4	7.4	7.5	7.4
SeqLoRA	44.6	32.7	53.7	43.7	2.3	0.6	1.9	1.6
IncLoRA	66	64.9	68.3	66.4	63.3	58.5	61.7	61.2
Data&Learning-Based Continual Learning								
Replay	55.2	56.9	61.3	57.8	55	54.6	53.1	54.2
EWC	48.7	47.7	54.5	50.3	45.3	44.5	45.6	45.1
LwF	54.4	53.1	49.6	52.3	50.1	43.1	47.4	46.9
Architecture-Based Continual Learning								
L2P	60.3	61.7	61.1	60.7	57.5	53.8	56.9	56.1
LFPT5	67.6	72.6	77.9	72.7	70.4	68.2	69.1	69.2
O-LoRA	75.4	75.7	76.3	75.8	72.3	64.8	71.6	69.6
LC-BL	76.9	76.5	76.8	76.7	68.4	67.3	71.8	69.2
AM-LoRA	78.1	**79.8**	76.2	78.0	72.7	**73.3**	71.8	**72.6**
Ours								
CLoRA	**79.7**	79.1	**78.2**	**79.0**	**73.6**	66.4	72.4	70.8
+ O-LoRA	76.6	75.5	75.4	75.8	72.0	67.3	**77.3**	72.2
Multi-Task Learning Ceilings								
PerTaskFT	70.0	70.0	70.0	70.0	78.1	78.1	78.1	78.1
MTL	80.0	80.0	80.0	80.0	76.5	76.5	76.5	76.5

Table 5: Results on two CL benchmarks with T5-large base model. Averaged accuracy after training on the last task is reported. **Bold** font indicates the highest performance across all compared CL methods.

with one epoch, with constant learning rate of 1e-3, batch size of 64, dropout rate of 0.1, weight decay rate of 0, and LoRA dim r of 8. CLoRA regularization matrix size k is set to 256.

5.2.2 Results and Analysis

We report the results in Table 5, observations and analysis are listed as follows.

For Standard CL Benchmark Results demonstrate that CLoRA outperforms all comparison methods, include the most related strong baseline O-LoRA, with a notable margin. We attribute this to the advantage of CLoRA toward O-LoRA: 1) CLoRA helps learning in the first finetuning stage while O-LoRA not; 2) CLoRA can independently alter k for balancing learning and forgetting, while "k" equivalent in O-LoRA is restrained by LoRA r.

When combining with O-LoRA, performance is competitive with O-LoRA, but underperforms default CLoRA setting. We attribute this to the common knowledge that contribute to both tasks during training, preventing LoRAs from share learned knowledge would lead to sub-optimal results. This is also demonstrated in the performance margin between PerTaskFT and MTL, and also results demonstrated by LC-BL and AM-LoRA.

For Large Number of Tasks Benchmark CLoRA outperforms strong baselines including O-LoRA, LFPT5, and LC-BL. When combined with O-LoRA, the "+ O-LoRA" achieves performance competitive with AM-LoRA. We attribute this superiority to CLoRA's effective mitigation of catastrophic forgetting on large number of tasks benchmark, especially when combined with O-LoRA.

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 both one-stage LLM finetuning and continual learning settings. Experiment results demonstrate the effectiveness of CLoRA as a 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.

7 Limitations

There are still several limitations that we reserve for future work: 1) We only use the simplest random initialization and the combining with O-LoRA for regularization matrix. Insight for more dedicated choice may benefit CLoRA learning, and we leave those to the future work. 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.

Acknowledgments

We would like to thank anonymous reviewers for their suggestions and comments sincerely. The work was supported by the Beijing Natural Science Foundation (L247010), and Super Computing Platform of Beijing University of Posts and Telecommunications.

References

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

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.

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.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors. 2016. *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*. Springer International Publishing, Cham.

Jialin Liu, Jianhua Wu, Jie Liu, and Yutai Duan. Learning attentional mixture of loras for language model continual learning. *Preprint*, arXiv:2409.19611.

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.

Fuli Qiao and Mehrdad Mahdavi. Learn more, but bother less: Parameter efficient continual learning.

Chengwei Qin and Shafiq Joty. 2022. LFPT5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. In *International Conference on Learning Representations*.

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.

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

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.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. Learning to prompt for continual learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 139–149, New Orleans, LA, USA. IEEE.

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.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

A Detailed Experiment Setups for One-Stage Finetuning

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[2], training is conducted using trainer in Huggingface transformers library[3], with DeepSpeed ZeRO(Rajbhandari et al., 2020) integration.

A.3 Additional CLoRA variants

We use the simplest random initialization for CLoRA regularization matrix in the main paper. Considering the idea of PiSSA and MiLoRA that explore 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 $U[:, \mathbf{s}]$, $P_B \in \mathbb{R}^{n\times k}$ as $V[:, \mathbf{s}]$, where $\mathbf{s}$ is a list of selecting index with length k. We add two CLoRA variants as follows, and conduct experiments on commonsense reasoning setting,

- **CLoRA-major:** Use SVD major components to initialize CLoRA regularization matrix.
- **CLoRA-minor:** Use SVD minor components to initialize CLoRA regularization matrix.

A.4 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. Results for LLaMA-3-8b are also included for CLoRA-random. All models use LoRA rank r of 32 unless specified otherwise.

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

[3] https://github.com/huggingface/transformers

A.4.1 Analysis for different CLoRA variants

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).

B Detailed Experiment Setups for Continual Learning

B.1 Dataset Details

We list the details of the datasets used in Table 9. Order of finetuning are listed in Table 10.

B.2 Computation Environment

All of our experiments are conducted on 1 NVIDIA GeForce RTX 3090 GPU. All methods for LoRA subsequents use Huggingface peft library, training is conducted using trainer in Huggingface transformers library, with DeepSpeed ZeRO integration.

B.3 Comparison Methods

Here we provide details for continual learning baselines for our continual learning experiment setting.

- **Replay:** data-based method that replay samples from old tasks when learning new tasks to avoid forgetting.
- **L2P:** architecture-based method that uses the input to dynamically select and update prompts from the prompt pool in an instance-wise fashion.
- **LFPT5:** architecture-based method that continuously train a soft prompt that simultaneously learns to solve the tasks and generate training samples for replay.
- **EWC:** learning-based method that finetune the whole model with a regularization loss that prevents updating parameters that could interfere with previously learned tasks.
- **LwF:** learning-based method that constrains the shared representation layer to be similar to its original state before learning the new task.
- **O-LoRA:** architecture and learning-based method that prevent subsequent LoRA update interfere previous.

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 reasoning (CS) and Math settings.

- **LC-BL:** improves O-LoRA by employing sensitivity-based analysis of low-rank matrix parameters.
- **AM-LoRA:** improves O-LoRA by introducing modeling structure that efficiently leverage the distinctive contributions of each LoRA.

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.

Dataset name	**Category**	**Task**	**Domain**
Yelp	CL Benchmark	sentiment analysis	Yelp reviews
Amazon	CL Benchmark	sentiment analysis	Amazon reviews
DBpedia	CL Benchmark	topic classification	Wikipedia
Yahoo	CL Benchmark	topic classification	Yahoo Q&A
AG News	CL Benchmark	topic classification	news
MNLI	GLUE	NLI	various
QQP	GLUE	paragraph detection	Quora
RTE	GLUE	NLI	news, Wikipedia
SST-2	GLUE	sentiment analysis	movie reviews
WiC	SuperGLUE	word sense disambiguation	lexical databases
CB	SuperGLUE	NLI	various
COPA	SuperGLUE	QA	blogs, encyclopedia
BoolQA	SuperGLUE	boolean QA	Wikipedia
MultiRC	SuperGLUE	QA	various
IMDB	SuperGLUE	sentiment analysis	movie reviews

Table 9: Summary of datasets used in the continual learning setting.

Order	Task Sequence
1	dbpedia → amazon → yahoo → ag
2	dbpedia → amazon → ag → yahoo
3	yahoo → amazon → ag → dbpedia
4	mnli → cb → wic → copa → qqp → boolqa → rte → imdb → yelp → amazon → sst-2 → dbpedia → ag → multirc → yahoo
5	multirc → boolqa → wic → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yelp → amazon → yahoo
6	yelp → amazon → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yahoo → multirc → boolqa → wic

Table 10: Order of finetuning in the continual learning setting.