# Can Spectral-Clipping Enable Better Learning While Forgetting Less for Low-Rank Adaptation?

**Hyowon Wi**[1] **and Noseong Park**[1]
[1]Korea Advanced Institute of Science and Technology (KAIST)
{hyowon.wi,noseong}@kaist.ac.kr

## Abstract

In recent years, low-rank adaptation (LoRA) has emerged as a significant paradigm that freezes pre-trained weights and introduces small, learnable adapters instead of fine-tuning the full set of parameters. In this work, we uncover several key insights regarding the *singular* components of network parameters based on Singular Value Decomposition (SVD). Firstly, the *principal* singular components with large singular values in pre-trained network parameters can be effectively reused during fine-tuning, whereas the *minor* components with smaller singular values are more task-specific and require substantial adaptation. Secondly, we first establish the theoretical connection that the uncontrolled growth of singular values in LoRA adapters leads to the forgetting of pre-trained knowledge — a well-known issue referred to as *catastrophic forgetting*. Building on these observations, we propose **SCLoRA**, which injects parameterized singular components with spectral clipping into the pre-trained model in a way that is aware of the spectral distribution of the pre-trained model. **SCLoRA** effectively adapts to new tasks by focusing updates on components that require adaptation, while simultaneously alleviating catastrophic forgetting. We conduct extensive experiments and demonstrate that **SCLoRA** not only improves downstream performance but also effectively retains pre-trained knowledge.

## 1 Introduction

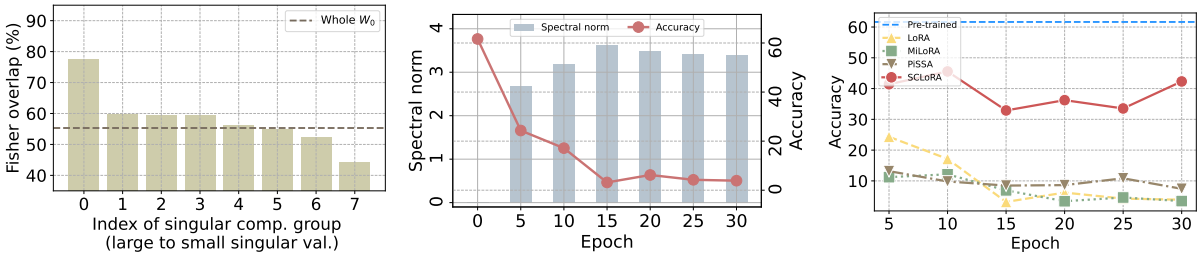
Pre-trained language models (PLMs) have achieved remarkable performance in various natural language processing tasks (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2019; He et al., 2020; Touvron et al., 2023a; Achiam et al., 2023; Anil et al., 2023). The common way to adapt pre-trained language models to downstream tasks is *fine-tuning*. However, fine-tuning all parameters and storing copies of the large model for each downstream task results in significant cost and memory consumption. To address this issue, recent studies suggest parameter-efficient fine-tuning (PEFT) methods (Hu et al., 2021; Zhang et al., 2023; **?**; Liu et al., 2024; Jiang et al., 2024; Meng et al., 2024; Wang et al., 2025), fine-tuning with only a small number of parameters. Low-Rank Adaptation (LoRA) (Hu et al., 2021), a representative PEFT approaches that keeps the pre-trained weights frozen and updates only a small number of parameters, has shown promising performance while being both storage- and compute-efficient.

In recent years, many studies have investigated the properties of *singular components* with Singular Value Decomposition (SVD) in LoRA (Zhang et al., 2023; Meng et al., 2024; Wang et al., 2025; Bałazy et al., 2024; Yang et al., 2024b; Wang et al., 2024). A singular component refers to a rank-1 matrix formed by the product of a pair of left and right singular vectors and their corresponding singular value. Specifically, a *principal* singular component refers to one associated with a larger singular value, representing the global structure of the matrix (Abdi and Williams, 2010; Meng et al., 2024). Conversely, a *minor* singular component corresponds to a smaller singular value and is often considered as noise (Wang et al., 2025). In deep learning, however, because learned weight matrices are typically full rank (Hu et al., 2021; Yu and Wu, 2023; Garg et al., 2025), the minor singular components are not merely noise; rather, they also encode detailed information within the matrix.

**Motivations.** Recent studies have explored SVD-based variants of LoRA by selecting and modifying specific singular components (Meng et al., 2024; Wang et al., 2025). These approaches assume that principal components contain important information and minor components are merely noise. However, a principled understanding of how different singular components contribute to task adaptation



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25708–25734




(a) Fisher overlap

(b) Trade-off between spectral norm and pre-trained accuracy

(c) Pre-trained accuracy trajectory

Figure 1: (a) The Fisher overlap (Kirkpatrick et al., 2017) between the pre-trained task (BookCorpus) and the fine-tuning task (MRPC), evaluated over partial reconstructions of the pre-trained network parameters obtained by grouping singular components sorted from large to small according to their singular values. Additional visualizations for other datasets are in Section E. (b) The trade-off between the spectral norm of the adapter and the accuracy on the pre-trained task (BookCorpus) during fine-tuning of LoRA on the STS-B dataset from the GLUE benchmark for RoBERTa$_{\text{base}}$. (c) Pre-trained accuracy trajectory across fine-tuning epochs with various baselines.

and knowledge preservation is still missing. To address this, we conduct a systematic analysis of LoRA through the lens of singular components. The first evidence is that the principal singular components of the pre-trained network parameters can be reused for the fine-tuning task to a great extent; the minor singular components become more task-specific and thus require a significant adaptation. To quantify spectrum-wise task alignment between pre-training and fine-tuning, we compute the *Fisher overlap* (Kirkpatrick et al., 2017; Yao and Hansen, 2022; Qian et al., 2024) on spectrally decomposed pre-trained parameters. Specifically, we apply SVD, sort singular values in descending order, and group the corresponding singular components. Fisher overlap is then computed on partial reconstructions from each group, enabling alignment to be analyzed across spectral scales. A higher Fisher overlap indicates stronger alignment, suggesting that the corresponding components are more transferable. The detailed formulation of the Fisher overlap is provided in Section E. Fig. 1 (a) shows that overlap gradually decreases as the singular components correspond to smaller singular values, suggesting that the principal singular components are already aligned, while minor singular components need more task-specific adaptation.

Beyond identifying where adaptation should occur, we further investigate how adaptation affects pretrained knowledge. Crucially, we establish the first theoretical connection showing that the growth of adapter singular values leads to the reduction of pretrained priors (see Theorem 3.1) during the fine-tuning optimization process. This reduction can result in a phenomenon called *catastrophic forgetting*, where the model rapidly forgets pre-trained knowledge during fine-tuning. Such forgetting undermines the scalability and reliability of pre-trained models, making it essential to address this issue (Wang et al., 2025; Yang et al., 2024b; Ren et al., 2024; Yang et al., 2024a; Dou et al., 2024). It is known that, however, during typical stochastic optimization, the spectral norm of weight matrices tends to grow rapidly (See Section 3.2 for details). Consequently, this norm growth leads to catastrophic forgetting in common fine-tuning settings. Empirically, Fig. 1(b) shows that LoRA exhibits a significant increase in the singular values of its adapter during fine-tuning. As in Fig. 1 (c), this increase is associated with performance degradation on the pre-trained task, indicating that LoRA is vulnerable to catastrophic forgetting. This phenomenon exposes a fundamental limitation of existing PEFT methods: unconstrained spectral growth results in significant forgetting.

**Main idea.** Motivated by these insights, we propose a **L**ow-**R**ank **A**daptation with **S**pectral **C**lipping, called **SCLoRA**, a new PEFT framework that directs adaptation through the injection of spectrally clipped singular components, preventing the uncontrolled spectral growth observed in conventional updates. **SCLoRA** constructs adapters as singular components through a parameterized SVD and constrains their singular values using a distribution-aware bound derived from the pre-trained spectral distribution. This constraint prevents excessive spectral growth and guides learning toward the relatively minor regions of the spec-

trum. As shown in our analysis, this aligns updates with the spectral components that genuinely require adaptation, enabling effective downstream learning. Moreover, since our theoretical results indicate that increasing singular values reduces the pretrained prior, bounding them according to the pretrained spectrum helps preserve the prior knowledge encoded during pre-training, thereby mitigating catastrophic forgetting. We conduct extensive experiments to evaluate the effectiveness of **SCLoRA**, demonstrating that it consistently outperforms LoRA and its variants across various tasks. Additionally, we assess catastrophic forgetting across multiple baseline models, showing that **SCLoRA** significantly mitigates the forgetting of pre-trained knowledge. Our key contributions can be summarized as follows:

- To our knowledge, we are the first to show that the minor singular components of network parameters require substantial adaptation, and we theoretically demonstrate that the growth of adapter singular values results in catastrophic forgetting.
- In Section 3.1, we propose **SCLoRA**, which injects the singular components with spectral clipping using parameterized SVD, ensuring that the pre-trained model efficiently adapts to new tasks and mitigates the catastrophic forgetting problem.
- In Section 5 and Section 6, extensive experiments demonstrate that **SCLoRA** efficiently adapts to the new task with performance gain and significant reduction in catastrophic forgetting across diverse tasks and model scales.

## 2 Preliminaries & Related Works

### 2.1 Low-Rank Adaptation and SVD

LoRA (Hu et al., 2021) suggests the low-rank update of the pre-trained weights by the product of two low-rank matrices. For $h = W_0 x$, the modified forward pass becomes:

$$h = W_0 x + \Delta W x = W_0 x + BAx, \quad (1)$$

where $W_0, \Delta W \in \mathbb{R}^{d_1 \times d_2}$, $A \in \mathbb{R}^{r \times d_2}$ and $B \in \mathbb{R}^{d_1 \times r}$ with $r \ll \{d_1, d_2\}$. $A$ is initialized with a random Gaussian initialization and $B$ with zero, so $\Delta W = BA$ is initially zero at the beginning of training. After fine-tuning, the learnable adapter $\Delta W$ can be integrated into the pre-trained weight $W$ without modifying the original model architecture or adding any additional inference overhead.

**SVD-based LoRA.** PiSSA (Meng et al., 2024) assumes that the principal components hold important information, decomposing the network parameters into principal and residual components using explicit SVD. Then the residual components are frozen, while the adapter is initialized with the principal components and directly updated. Conversely, MiLoRA (Wang et al., 2025) proposes directly modifying the minor components of the pre-trained networks, assuming they are noisy and less important, in order to better preserve the pre-trained knowledge. CorDA (Yang et al., 2024b) performs SVD on $W_0 C$ with input activation covariance $C$ from a small calibration set, to obtain a context-aligned low-rank decomposition. LoRA-GA (Wang et al., 2024) computes full-parameter gradients on a small sample set and applies SVD to obtain a low-rank subspace, initializing LoRA adapters so that their updates approximate the full-gradient direction. Unlike existing initialization-based method, AdaLoRA (Zhang et al., 2023) adopt parameterized SVD to dynamically adjusts the rank for LoRA layer based on a sensitivity-driven importance score. They focus on pruning the number of ranks to meet a predefined budget using heuristic importance scores. Despite their architectural differences, however, existing methods do not explicitly provide a principled way to determine where task-specific adaptation should occur or how pretrained knowledge should be preserved, nor do they control the spectral dynamics of the updates that govern this process.

### 2.2 Catastrophic forgetting and LoRA

Catastrophic forgetting refers a well-known issue in the field of deep learning (McCloskey and Cohen, 1989; French, 1999; Kirkpatrick et al., 2017), where the models forget previously acquired knowledge during adaptation to new tasks. To address this challenge, recent studies have proposed various approaches, including knowledge distillation (Li and Hoiem, 2017; Hou et al., 2019), rehearsal (Riemer et al., 2018; Yang et al., 2023) and dynamic architectures (Yan et al., 2021). This issue is particularly severe in LLMs, which learn extensive world knowledge through the pre-training process on massive datasets. During the fine-tuning process, where task-specific information is learned based on this world knowledge, forgetting

the pre-trained knowledge can significantly undermine the stability and scalability of the models. Catastrophic forgetting has also been observed in parameter-efficient fine-tuning methods, including LoRA, prompting recent studies to propose various approaches to mitigate this issue (Wang et al., 2025; Yang et al., 2024b; Ren et al., 2024; Yang et al., 2024a; Dou et al., 2024).

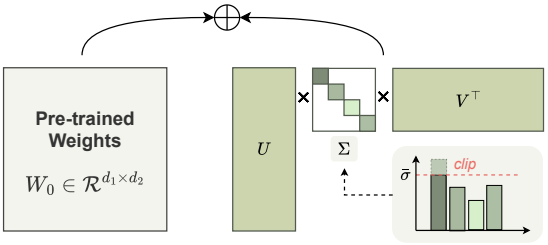



Figure 2: The architecture of **SCLoRA**

## 3 Proposed Method

### 3.1 Low-Rank Adaptation with Spectral Clipping

Our analysis shows that the minor singular components of the pre-trained weights exhibit a lower degree of task alignment and therefore require substantially more task-specific adaptation than the principal components. Furthermore, we empirically and theoretically observe that the growth of adapter singular values during fine-tuning weakens the pre-trained prior and consequently inducing catastrophic forgetting. Therefore, existing SVD-based approaches such as (Meng et al., 2024; Wang et al., 2025, 2024; Yang et al., 2024b; Zhang et al., 2023) operate in the spectral domain by initializing adapters with SVD-derived components at the weight, activation, or gradient level, or by controlling the number of rank via parameterized SVD; however, the subsequent adaptation dynamics along the singular spectrum remain uncontrolled. Our goal is to ensure that, regardless of the initialization scheme, the adapter primarily learns within the minor spectral subspace of the pre-trained weights—where task-specific adaptation is needed—while avoiding distortion of the principal components, and our approach explicitly controls the spectrum during fine-tuning.

To realize this objective, we introduce spectral clipping on the adapter singular values. Singular Value Clipping (SVC) has been widely applied in various domains, including GANs and CNNs (Saito et al., 2017; Senderovich et al., 2022). By controlling the spectral norm of each layer, SVC preserves the Lipschitz continuity of the network, leading to improved training stability, mitigation of exploding and vanishing gradients, and enhanced generalization and robustness. However, applying explicit SVD on adapter and clipping at every training step is computationally infeasible for LLM-scale parameters. We therefore adopt a parameterized SVD framework, which enables learning directly in the spectral domain while allowing the singular values of the adapter to be clipped within a controlled range in an efficient manner. Formally, we parameterize the adapter as a low-rank matrix whose singular values are clipped with respect to an upper bound derived from the spectral distribution of the pre-trained weights.

$$W = W_0 + \Delta W = W_0 + U\Sigma V^\top, \qquad (2)$$

where $U \in \mathbb{R}^{d_1 \times r}$, $V \in \mathbb{R}^{d_2 \times r}$ are parameterized left and right singular vectors, respectively, and $\Sigma \in \mathbb{R}^r$ contains the parameterized singular values $\{s_n\}_{1 \le n \le r}$. At the start of fine-tuning, $U, V$ are initialized with random $r$ singular vectors of $W_0$, or $U$ is initialized with zero and $V$ with a random Gaussian. Note that SVD on $W_0$ is performed only once before fine-tuning, and the actual operation does not involve any explicit decomposition or reconstruction of $W_0$ during the fine-tuning process.

A natural question that follows is: *how should we select an appropriate upper bound on the singular values?* As illustrated in Fig. 1 (a), the principal components of the pretrained network are already aligned with the new task, whereas the minor components exhibit a lower degree of task alignment and therefore require more substantial adaptation. This observation suggests introducing spectral-aware constraints that guide learning toward the regions of the spectrum where adaptation is needed. However, using a fixed scalar upper bound is generally arbitrary and sensitive to model scale and layer dimensionality. Instead, defining the upper bound in terms of the quantiles of the pretrained spectral distribution provides a scale-independent and distribution-aware constraint. Accordingly, we impose a constraint that limits the growth of the adapter singular values while respecting the spectral structure of the pretrained weights. We realize this constraint through *quantile*-based clipping, which restricts the adapter updates to remain within a bounded region defined by the pretrained spectral distribution. Formally, we denote the singular values of the pretrained weight matrix

$W_0$ as $\sigma_i(W_0)_{i=1}^d$. The spectral bound $\bar{\sigma}$ is then defined as the $q$-th quantile $Q_q(\cdot)$ of the pretrained spectral distribution:

$$\bar{\sigma} = Q_q(\sigma_i(W_0)_{i=1}^d). \tag{3}$$

Then, the parameterized singular values are constrained by quantile-based spectral clipping as:

$$s_n \leftarrow \min\big(\max(s_n, 0), \bar{\sigma}\big), \tag{4}$$

for $n \in \{1, \ldots, r\}$. This formulation explicitly links the spectral bound of the adapter to the pretrained spectrum while remaining independent of the absolute parameter scale. Thus, quantile-based spectral clipping establishes a distribution-aware trust region, enabling controlled adaptation in the regions that require greater modification. An ablation study on the effect of $q$ is in Section 7.1. To ensure that the singular values remain non-negative, we set the lower bound to zero in accordance with the definition of SVD. Furthermore, to enforce the orthogonality of the singular vectors, i.e., $U^\top U = VV^\top = I$, we apply the following regularization term:

$$R(U, V) = \|U^\top U - I\| + \|VV^\top - I\| \tag{5}$$

where $I \in \mathbb{R}^{r \times r}$ indicates an identity matrix. This regularization term is controlled by the orthogonal regularization coefficient $\gamma$. We verify the orthogonality of the parameterized singular vectors in Section Q. We present the training process in Algorithm 1 of Section J.

### 3.2 Theoretical analysis

The optimization of deep learning models, including LoRA, can be interpreted as a Maximum A Posteriori (MAP) estimation of the network parameters $\theta$ on the training data. In transfer learning, the model is first pre-trained on the dataset $\mathcal{D}_A$ and then fine-tuned on the dataset $\mathcal{D}_B$. Following the standard Bayesian formulation of MAP estimation, the posterior to be maximized in the MAP estimation is formulated as:

$$\begin{aligned} p(\theta|\mathcal{D}_A, \mathcal{D}_B) &= \frac{p(\mathcal{D}_B|\theta, \mathcal{D}_A)p(\theta|\mathcal{D}_A)}{p(\mathcal{D}_B|\mathcal{D}_A)} \\ &= \frac{p(\mathcal{D}_B|\theta)p(\theta|\mathcal{D}_A)}{p(\mathcal{D}_B|\mathcal{D}_A)}. \end{aligned} \tag{6}$$

Taking a logarithm of the posterior, the MAP objective becomes:

$$\begin{aligned} \theta^* &= \operatorname*{argmax}_\theta \log p(\theta|\mathcal{D}_A, \mathcal{D}_B) \\ &= \operatorname*{argmax}_\theta \big[\log p(\mathcal{D}_B|\theta) + \log p(\theta|\mathcal{D}_A)\big]. \end{aligned} \tag{7}$$

The first term corresponds to the likelihood of $\mathcal{D}_B$ and is expressed as the loss function for the fine-tuning task, while the second term represents the prior over the parameters induced by $\mathcal{D}_A$. During fine-tuning, we treat the posterior from the pre-trained task, $p(\theta|\mathcal{D}_A)$, as the prior for the fine-tuning task. However, since the true posterior is intractable, it can be expressed as a function $f(\theta)$ and approximated around the pre-trained mode $\theta_0$ using the Laplace approximation, a well-established method in Bayesian deep learning for handling intractable posteriors (Kirkpatrick et al., 2017; Ritter et al., 2018; Wang et al., 2021; Matena and Raffel, 2022; Gawlikowski et al., 2023) as:

$$\log \hat{p}(\theta|\mathcal{D}_A) = f(\theta_0) - \frac{1}{2}(\theta - \theta_0)^\top F(\theta - \theta_0), \tag{8}$$

where $F$ denotes the Fisher information matrix. The following theorem shows that this approximated prior is upper-bounded by the singular values of the difference between the pre-trained and fine-tuned parameters.

**Theorem 3.1.** *Under the Laplace approximation, the approximated log prior probability satisfies:*

$$\log \hat{p}(\theta \mid \mathcal{D}_A) \leq f(\theta_0) - \lambda_{\min}(F)\sqrt{\sum_{n=1}^{r} \sigma_n^2}, \tag{9}$$

*where $\lambda_{\min}(F)$ denotes the smallest eigenvalue of $F$, and $\sigma_n$ is the $n$-th singular value of $\theta - \theta_0$.*

The proof is described in Section F. It is worth noting that the negligibility of the higher–order terms in the Laplace approximation is supported by a well-established body of theoretical results in Bayesian asymptotics (Kass et al., 1990), where the omitted error term vanishes as the posterior concentrates around $\theta_0$. We provide further discussion of this approximation error in Section G.

According to Theorem 3.1, $\log \hat{p}(\theta|\mathcal{D}_A)$ is upper bounded by the singular values of the parameter difference, which is adapter in LoRA. Specifically, larger singular values of the adapter lead to a decrease in the posterior from the pre-training task, resulting in the loss of pre-trained knowledge, the phenomenon called *catastrophic forgetting*. The catastrophic forgetting problem undermines the strengths of pre-trained models and hinders their adaptability to new tasks, making it crucial to address in order to maintain scalability and reliability in transfer learning (Wang et al., 2025; Yang et al.,

2024b; Ren et al., 2024; Yang et al., 2024a; Dou et al., 2024). In stochastic optimization, however, the spectral norm of weight matrices tends to grow rapidly. The following proposition from Zhai et al. (2023) establishes a lower bound on the spectral norm of the ideal update.

**Proposition 3.2.** *From Zhai et al. (2023), it holds:*

$$\|\Delta\| \geq \sqrt{d}\sqrt{1 - \frac{1}{d^2}\sum_{i,j=1}^{d}\frac{\omega_{i,j}^2}{\mu^2 i, j + \omega_{i,j}^2}}. \quad (10)$$

The noise second moment $\omega^2$ is typically in the order of $\mu^2$. Hence, Theorem 3.2 indicates that the spectral norm of the ideal update should be large, growing linearly with $\sqrt{d}$. Moreover, for large batch sizes we would have $\omega^2 \ll 1$, resulting in $\|\Delta\| \sim \sqrt{d}$. Proposition 3.2 from Zhai et al. (2023) demonstrates that this growth becomes more pronounced in high-dimensional settings when adaptive optimizers are used. Therefore, in general probabilistic optimization for transfer learning, including LoRA, the spectral norm is implicitly driven toward larger values during fine-tuning. This increase in the largest singular value reduces the posterior probability of the pre-trained knowledge under the MAP formulation, thereby leading to catastrophic forgetting. Empirically, a consistent spectrum–forgetting trend is observed in our experiments: In Section N, the singular values of the LoRA adapter grow during fine-tuning and this growth is accompanied by degradation of the pre-training performance, indicating that LoRA can be vulnerable to catastrophic forgetting in practice.

## 4 Complexity analysis

In this section, we analyze the complexity of SCLoRA. Empirical runtime and GPU usage are further detailed in Section R.

**Parameter Complexity.** LoRA updates $(d_1 + d_2)r$ parameters per layer. **SCLoRA** introduces a minimal overhead of only $r$ parameters for the singular values, resulting in $(d_1+d_2)r+r$ trainable parameters. This successfully preserves the inherent parameter efficiency of the baseline method.

**Computational Complexity.** During training, standard LoRA requires $\mathcal{O}(d_1 d_2 r)$ operations. **SCLoRA** operates with a slightly higher complexity of $\mathcal{O}(d_1 d_2 r + (d_1 + d_2)r + r^3)$, consistent with other parameterized SVD-based approaches. Due to the strict low-rank constraint, this additional overhead is mathematically negligible. During inference, standard LoRA scales as $\mathcal{O}(d_1 d_2 r)$. By scaling the singular vectors with parameterized singular values, **SCLoRA** achieves an inference complexity of $\mathcal{O}(d_1 d_2 r + d_2 r)$, which remains highly comparable to the baseline.

## 5 Experiments

We empirically verify that **SCLoRA** efficiently adapts to the task and improves the performance.

### 5.1 Natural Language Understanding

**Experimental setup.** Following Hu et al. (2021) and Zhang et al. (2023), we fine-tune the pre-trained RoBERTa$_{\text{base}}$ and DeBERTa$_{\text{base}}$ models on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018a), setting the rank to $r = 8$ for RoBERTa$_{\text{base}}$ and $r \in \{2, 8\}$ for DeBERTa$_{\text{base}}$. We report the Matthews correlation for CoLA, Spearman's correlation for STS-B, and accuracy for all other tasks. Detailed descriptions are provided in Section M.1.2.

**Main results.** In Table 1, **SCLoRA** outperforms various baselines on average. In particular, MiLoRA, which is closely related to our method, fails to achieve optimal performance due to information loss and uncontrolled fine-tuning. By contrast, **SCLoRA** efficiently adapts to the new task by injecting spectrally clipped singular components.

### 5.2 Question Answering

**Experimental setup.** Following Zhang et al. (2023), we fine-tune DeBERTaV3$_{\text{base}}$ (He et al., 2021) under four different budgets: 0.08%, 0.16%, 0.32%, and 0.65% of the total pre-trained parameters. We evaluate **SCLoRA** on two question-answering (QA) benchmarks—SQuAD v1.1 (Rajpurkar, 2016) and SQuAD v2.0 (Rajpurkar et al., 2018)—using the Exact Match (EM) and F1 metrics. Detailed descriptions are in Section M.2.2.

**Main results.** Table 2 illustrates that **SCLoRA** delivers performance comparable to or better than existing baselines in most parameter budgets, suggesting that spectral clipping of the adapter is effective for QA adaptation under various budgets. Furthermore, it outperforms strong baselines like AdaLoRA in most cases, demonstrating the robustness of our spectral constraint approach.

| Method | # Params | MNLI | SST-2 | CoLA | QQP | QNLI | RTE | MRPC | STS-B | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| *Model: RoBERTa$_{base}$* | | | | | | | | | | |
| LoRA | 1.33M | 87.93 | 94.80 | 64.49 | 90.94 | 92.73 | 80.39 | 89.05 | 90.87 | 86.40 |
| AdaLoRA | 1.27M | 87.90 | 94.80 | 62.41 | 90.16 | 92.67 | 81.59 | 89.38 | 91.08 | 86.25 |
| PiSSA | 1.33M | **87.95** | 94.53 | 64.66 | 90.97 | 92.53 | 79.18 | 89.79 | 90.96 | 86.32 |
| MiLoRA | 1.33M | 87.88 | 94.69 | 64.31 | **91.02** | 92.96 | 81.35 | 89.30 | 90.96 | 86.56 |
| LoRA+ | 1.33M | 86.96 | 93.92 | 63.32 | 90.69 | 92.77 | 81.59 | 88.97 | 90.84 | 86.13 |
| LoRA-GA | 1.33M | 85.18 | 93.16 | 62.09 | 88.57 | 91.64 | 76.77 | 89.54 | 90.87 | 84.73 |
| CorDA | 1.33M | 86.28 | 94.38 | 62.43 | 90.14 | 92.36 | 78.10 | 89.87 | 90.68 | 85.53 |
| DoRA | 1.41M | 87.81 | 95.11 | 64.23 | 90.65 | 92.93 | 81.35 | 89.54 | 91.01 | 86.58 |
| **SCLoRA** | 1.33M | **87.95** | **95.37** | **64.79** | 90.87 | **93.09** | **83.15** | **90.32** | **91.22** | **87.08** |
| *Model: DeBERTaV3$_{base}$* | | | | | | | | | | |
| BitFit | 0.10M | 89.37 | 94.84 | 66.96 | 88.41 | 92.24 | 78.70 | 87.75 | 91.35 | 86.02 |
| HAdapter | 1.22M | 90.13 | 95.53 | 68.64 | 91.91 | 94.11 | 84.48 | 89.95 | 91.48 | 88.12 |
| PAdapter | 1.18M | 90.33 | 95.61 | 68.77 | 92.04 | 94.29 | 85.20 | 89.46 | 91.54 | 88.24 |
| LoRA$_{r=8}$ | 1.33M | 90.65 | 94.95 | 69.82 | 91.99 | 93.87 | 85.20 | 89.95 | 91.60 | 88.34 |
| AdaLoRA | 1.27M | **90.76** | 96.10 | 71.45 | 92.23 | 94.55 | 88.09 | 90.69 | 91.84 | 89.31 |
| **SCLoRA** | 1.33M | 90.36 | **96.33** | **71.49** | **92.33** | **94.57** | **89.41** | **91.58** | **92.19** | **89.78** |
| HAdapter | 0.61M | 90.12 | 95.30 | 67.87 | 91.65 | 93.76 | 85.56 | 89.22 | 91.30 | 87.93 |
| PAdapter | 0.60M | 90.15 | 95.53 | 69.48 | 91.62 | 93.98 | 84.12 | 89.22 | 91.52 | 88.04 |
| HAdapter | 0.31M | 90.10 | 95.41 | 67.65 | 91.54 | 93.52 | 83.39 | 89.25 | 91.31 | 87.60 |
| PAdapter | 0.30M | 89.89 | 94.72 | 69.06 | 91.40 | 93.87 | 84.48 | 89.71 | 91.38 | 87.90 |
| LoRA$_{r=2}$ | 0.33M | 90.30 | 94.95 | 68.71 | 91.61 | 94.03 | 85.56 | 89.71 | 91.68 | 88.15 |
| AdaLoRA | 0.32M | **90.66** | 95.80 | 70.04 | 91.78 | 94.49 | 87.36 | 90.44 | 91.63 | 88.86 |
| **SCLoRA** | 0.33M | **90.66** | **96.18** | **71.83** | **91.82** | **94.50** | **89.89** | **91.83** | **92.00** | **89.84** |

Table 1: Comparison of various methods with RoBERTa$_{base}$ and DeBERTaV3$_{base}$ on GLUE tasks with different random seeds. Results with standard deviations are in Section M.1.4.

| Method | SQuADv1.1 | | | | SQuADv2.0 | | | |
|---|---|---|---|---|---|---|---|---|
| | 0.08% | 0.16% | 0.32% | 0.65% | 0.08% | 0.16% | 0.32% | 0.65% |
| Full FT | 86.0 / 92.7 | | | | 85.4 / 88.4 | | | |
| HAdapter | 84.4/91.5 | 85.3/92.1 | 86.1/92.7 | 86.7/92.9 | 83.4/86.6 | 84.3/87.3 | 84.9/87.9 | 85.4/88.3 |
| PAdapter | 84.4/91.7 | 85.9/92.5 | 86.2/92.8 | 86.6/93.0 | 84.2/87.2 | 84.5/87.6 | 84.9/87.8 | 84.5/87.5 |
| LoRA | 86.4/92.8 | 86.6/92.9 | 86.7/93.1 | 86.7/93.1 | 84.7/87.5 | 83.6/86.7 | 84.5/87.4 | 85.0/88.0 |
| AdaLoRA | 87.2/93.4 | 87.5/93.6 | 87.5/93.7 | 87.6/93.7 | **85.6/88.7** | 85.7/88.8 | 85.5/88.6 | 86.0/88.9 |
| **SCLoRA** | **87.6/93.6** | **88.1/93.9** | **88.2/94.1** | **88.6/94.3** | 85.3/88.3 | **86.0/89.0** | **86.0/88.8** | **86.2/89.0** |

Table 2: Comparison of various methods with DeBERTaV3$_{base}$ on SQuAD datasets

### 5.3 Commonsense reasoning

**Experimental setup.** We evaluate **SCLoRA** on the commonsense reasoning tasks. Following Hu et al. (2023), we amalgamate the training datasets from all 8 tasks to create the final training dataset and evaluate with individual testing for each task. We fine-tune LLaMA-7B (Touvron et al., 2023a) and LLaMA2-7B (Touvron et al., 2023b). The detailed descriptions are provided in Section M.3.2.

**Main results.** Table 3 summarizes the comparison of various fine-tuning methods on commonsense reasoning benchmarks. **SCLoRA** demonstrates superior performance across both LLaMA-7B and LLaMA2-7B architectures with average accuracy of 79.4 and 81.0. Importantly, SCLoRA not only yields substantial improvements over standard LoRA (+4.7% and +3.4% on average), but it also outperforms recent advanced methods such as DoRA and MiLoRA. These results highlights the effectiveness of spectral clipping on adapters for reasoning performance even in larger models.

## 6 Mitigation of catastrophic forgetting

This section validates that **SCLoRA** successfully mitigates *catastrophic forgetting*. Table 4 reports the spectral norm of adapter, fine-tuning accuracy (↑), and pre-trained task performance measured by accuracy (↑) or perplexity (↓), providing a quantitative comparison of catastrophic forgetting across different methods.

The table on the left shows the results of fine-tuning the RoBERTa$_{base}$ model using the MRPC and SST-2 datasets, respectively. After fine-tuning, the model was evaluated on the pre-

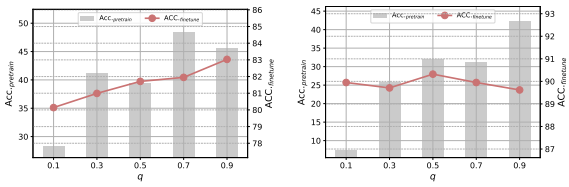
| Model | Method | #Params (%) | BoolQ | PIQA | SIQA | Hella. | Wino. | ARC-e | ARC-c | OBQA | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ChatGPT | - | - | 73.1 | 85.4 | 68.5 | 78.5 | 66.1 | 89.8 | 79.9 | 74.8 | 77.0 |
| LLaMA-7B | Prefix | 0.11 | 64.3 | 76.8 | 73.9 | 42.1 | 72.1 | 72.9 | 54.0 | 60.6 | 64.6 |
| | LoRA | 0.83 | 68.9 | 80.7 | 77.4 | 78.1 | 78.8 | 77.8 | 61.3 | 74.8 | 74.7 |
| | DoRA† | 0.43 | 70.0 | 82.6 | 79.7 | 83.2 | 80.6 | 80.6 | 65.4 | 77.6 | 77.5 |
| | DoRA | 0.84 | 69.7 | 83.4 | 78.6 | 87.2 | 81.0 | 81.9 | 66.2 | 79.2 | 78.4 |
| | **SCLoRA** | 0.83 | 70.9 | 83.8 | 80.1 | 88.3 | 81.5 | 82.8 | 66.5 | 81.0 | **79.4** |
| LLaMA2-7B | LoRA | 0.83 | 69.8 | 79.9 | 79.5 | 83.6 | 82.6 | 79.8 | 64.7 | 81.0 | 77.6 |
| | PiSSA | 0.83 | 67.6 | 78.1 | 78.4 | 76.6 | 78.0 | 75.8 | 60.2 | 75.6 | 73.8 |
| | MiLoRA | 0.83 | 67.6 | 83.8 | 80.1 | 88.2 | 82.0 | 82.8 | 68.8 | 80.6 | 79.2 |
| | DoRA† | 0.43 | 72.0 | 83.1 | 79.9 | 89.1 | 83.0 | 84.5 | 71.0 | 81.2 | 80.5 |
| | DoRA | 0.84 | 71.8 | 83.7 | 76.0 | 89.1 | 82.6 | 83.7 | 68.2 | 82.4 | 79.7 |
| | **SCLoRA** | 0.83 | 69.3 | 84.2 | 80.4 | 87.1 | 85.4 | 85.6 | 72.7 | 83.6 | **81.0** |

Table 3: Comparison of various methods with LLaMA on commonsense reasoning tasks

| *Model: RoBERTa$_{base}$* | | | | | | | | *Fine-tuning tasks: Commonsense reasoning* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Task | Method | MRPC | | | SST-2 | | | Task | Method | LLaMA-7B | | | LLaMA-2-7B | | |
| | | $\|\cdot\|_2$ | Acc.$_{ft}$ | Acc.$_{pt}$ | $\|\cdot\|_2$ | Acc.$_{ft}$ | Acc.$_{pt}$ | | | $\|\cdot\|_2$ | Acc.$_{ft}$ | PPL.$_{pt}$ | $\|\cdot\|_2$ | Acc.$_{ft}$ | PPL.$_{pt}$ |
| BC | Pre-trained | - | - | 61.64 | - | - | 61.64 | PG19 | Pre-trained | | - | 6.66 | | - | 6.61 |
| | LoRA | 3.39 | 89.05 | 3.77 | 12.12 | 94.80 | 32.35 | | LoRA | 13.16 | 74.7 | 8.69 | 21.46 | 77.6 | 11.36 |
| | **SCLoRA** | 0.94 | **90.32** | **32.00** | 0.45 | **95.37** | **51.29** | | **SCLoRA** | 5.09 | **79.4** | **7.79** | 8.70 | **81.0** | **8.29** |
| OWT | Pre-trained | | - | 68.21 | | - | 68.21 | C4$_{en}$ | Pre-trained | | - | 7.58 | | - | 7.61 |
| | LoRA | 3.39 | 89.05 | 18.36 | 12.12 | 94.80 | 54.33 | | LoRA | 13.16 | 74.7 | 10.54 | 21.46 | 77.6 | 15.26 |
| | **SCLoRA** | 0.94 | **90.32** | **47.27** | 0.45 | **95.37** | **65.67** | | **SCLoRA** | 5.09 | **79.4** | **9.53** | 8.70 | **81.0** | **10.71** |

Table 4: Comparison on catastrophic forgetting

trained datasets BookCorpus (BC) and OpenWebText (OWT). When applying LoRA to the MRPC, the downstream performance is 89.05; however, the spectral norm increases to 3.39, and the accuracy on the BC drops from 61.64 to 3.77. Similarly, the accuracy on the OWT drops substantially from 68.21 to 18.36. In contrast, under the same conditions, the spectral norm of **SCLoRA** is 0.94, which is approximately 27% of that of LoRA, while achieving accuracies of 32.00 and 47.27 on the BC and OWT, respectively. Similarly, in the right table, we evaluate LLaMA-7B and LLaMA2-7B fine-tuned on commonsense reasoning tasks, and report perplexity on PG19 and C4$_{en}$ datasets. Compared to LoRA, **SCLoRA** suppresses the increase in the spectral norm to approximately 40% while mitigating the increase in perplexity on the pre-training corpus, successfully preserving the knowledge of the original model. Overall, these results support that uncontrolled spectral norm growth is a key factor underlying catastrophic forgetting, and that constraining adaptation by injecting singular components with spectral clipping during fine-tuning enables stable task adaptation while effectively preserving pre-trained knowledge. Additional experiments, including the fine-tuning evolution of catastrophic forgetting and comparisons with additional baselines, are reported in Section O.



(a) RTE (b) MRPC

Figure 3: Sensitivity on $q$

## 7 Additional Studies

In Section P, we conduct sensitivity analyses of $\gamma$, and $r$, and ablation studies of layer-wise $q$ selection and automatic $\bar{\sigma}$.

### 7.1 Sensitivity study on $q$

In Fig. 3, we conduct a sensitivity analysis of $q$, which represents the $q$-th quantile, measuring performance changes on both the pre-training and fine-tuning tasks. When $q = 0.1$, the pre-training performance drops, which is consistent with Theorem 3.1 showing that excessively large adapter singular values weaken the pretrained prior and induce catastrophic forgetting. In contrast, once the quantile exceeds a moderate range, the pretrained performance rapidly recovers, indicating that spectral clipping is sufficient to preserve pretrained knowledge. Mean-

| Model | MRPC | | SST-2 | |
|---|---|---|---|---|
| | $\text{Acc.}_{\text{ft}}$ | $\text{Acc.}_{\text{pt}}$ | $\text{Acc.}_{\text{ft}}$ | $\text{Acc.}_{\text{pt}}$ |
| Pre-trained | - | 61.64 | - | 61.64 |
| LoRA | 89.05 | 3.77 | 94.81 | 32.35 |
| $\text{LoRA}_{UV^\top}$ | 89.22 | 3.12 | 94.75 | 39.39 |
| $\text{LoRA}_{\text{SVD}}$ | 89.62 | 17.57 | 95.03 | 49.65 |
| $\text{LoRA}_{\gamma=0}$ | 90.20 | 28.31 | 94.84 | 45.74 |
| **SCLoRA** | **90.32** | **32.00** | **95.37** | **51.29** |

Table 5: Ablation on the injected components

while, the fine-tuning performance is less sensitive to the choice of $q$. For example, in MRPC, we observe a slight improvement at $q = 0.1$, suggesting that weak spectral constraints can amplify task-specific signals in low-resource settings. However, this gain comes at the cost of severe degradation in pre-training performance and is therefore unlikely to be desirable in practice. Overall, we observe that when $q$ is set to a moderate range (e.g., $q \geq 0.3$), both pretrained retention and downstream performance remain stable. This indicates that extremely small quantile values should be avoided, whereas a broad moderate region yields consistently robust behavior across tasks.

### 7.2 Ablation study on the injected components

To analyze the influence of the injected components in **SCLoRA** on the performance of both pre-trained and fine-tuned knowledge, we conduct an ablation study on the following variants: i) 'LoRA' refers to the traditional LoRA; ii) '$\text{LoRA}_{UV^\top}$' applies orthogonal regularization to the singular vectors without the singular values; iii) '$\text{LoRA}_{\text{SVD}}$' initializes the singular values as ones, allowing them to be learnable from $\text{LoRA}_{UV^\top}$; iv) '$\text{LoRA}_{\gamma=0}$ applies singular value clipping without orthonormal regularization; and v) '**SCLoRA**' refers to the proposed method. We measure the accuracy on both the fine-tuning tasks with MRPC and SST-2, and the pre-trained task with BookCorpus dataset. As reported in Table 5, LoRA significantly sacrifices pre-training performance to adapt on new task. For the MRPC, accuracy on the pre-trained task drops from 61.64 to 3.77, while it achieves comparable accuracy on the fine-tuning task. $\text{LoRA}_{UV^\top}$ has limited expressiveness since its singular values are fixed, sacrificing either performance of pre-trained or fine-tuning task. $\text{LoRA}_{\text{SVD}}$ performs better due to its learnable singular values than $\text{LoRA}_{UV^\top}$. $\text{LoRA}_{\gamma=0}$ shows promising performance; however, since it does not ensure the orthogonality of the singular vectors, Theorem 3.1 cannot be applied, and thus it lacks a theoretical guarantee that catastrophic forgetting is consistently mitigated. Notably, **SCLoRA** learns the singular components with spectral clipping, ensuring both effective adaptation to the fine-tuning task and retention of pre-trained knowledge. For both datasets, **SCLoRA** achieves the best performance on both tasks.

## 8 Conclusion

We propose a novel LoRA method, motivated by the following two rigorous analyses regarding the singular components of network parameters: i) We, for the first time, analyze the Fisher overlap of the pre-trained weights across the singular components. Specifically, the principal singular components of pre-trained weights can be reused for fine-tuning tasks, whereas the minor singular components require significant adaptation; ii) The growth of singular values in the adapters directly causes the phenomenon *catastrophic forgetting*. From these analyses, we propose **SCLoRA**, which injects the parameterized singular components with spectral clipping. Comprehensive experiments show that **SCLoRA** achieves strong performance on fine-tuning tasks and successfully retains the pre-trained knowledge. For future direction, the spectrum clipping of SCLoRA could be extended to the Mixture of Experts (MoE) architecture.

## Limitation

Although $q$ is introduced as a hyperparameter, the sensitivity analysis in Section 7.1 shows that once $q$ exceeds a moderate range, both fine-tuning and pre-trained performance remain stable. While small variations on performance may exist, the method is generally robust to the choice of $q$. Moreover, this stability is observed consistently across multiple tasks and model scales, suggesting the practical sensitivity of $q$ is limited.

## Acknowledgments

Noseong Park is the corresponding author. This work was partly supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korean government (MSIT) (No. RS-202400457882, AI Research Hub Project, 34%; N10260110, AI Meta-Scientist, 33%), Samsung Electronics Co., Ltd. (No. G01240136, KAIST Semiconductor Research Fund (2nd), 33%).

## References


Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, and 1 others. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Klaudia Bałazy, Mohammadreza Banaei, Karl Aberer, and Jacek Tabor. 2024. Lora-xs: Low-rank adaptation with extremely small number of parameters. *arXiv preprint arXiv:2405.17604*.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. *TAC*, 7:8.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, and 1 others. 2024. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*.

Blair Bilodeau, Yanbo Tang, and Alex Stringer. 2023. On the tightness of the laplace approximation for statistical inference. *Statistics & Probability Letters*, 198:109839.

Shubhankar Borse, Shreya Kadambi, Nilesh Pandey, Kartikeya Bhardwaj, Viswanath Ganapathy, Sweta Priyadarshi, Risheek Garrepalli, Rafael Esteves, Munawar Hayat, and Fatih Porikli. 2024. Foura: Fourier low-rank adaptation. *Advances in Neural Information Processing Systems*, 37:71504–71539.

William L Briggs and Van Emden Henson. 1995. *The DFT: an owner's manual for the discrete Fourier transform*. SIAM.

Yang Cao. 2024. Sorsa: Singular values and orthonormal regularized singular vectors adaptation of large language models. *arXiv preprint arXiv:2409.00055*.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2018. Quora question pairs.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, and 1 others. 2024. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. 2024. Parameter-efficient fine-tuning with discrete fourier transform. *arXiv preprint arXiv:2405.03003*.

Isha Garg, Christian Koguchi, Eshan Verma, and Daniel Ulbricht. 2025. Revealing the utilized rank of subspaces of learning in neural networks. In *Proceedings of the AAAI Symposium Series*, volume 5, pages 151–158.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, and 1 others. 2023. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

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

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. 2019. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 831–839.

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. *arXiv preprint arXiv:2106.09685*.

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.

Ting Jiang, Shaohan Huang, Shengyue Luo, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and 1 others. 2024. Mora: High-rank updating for parameter-efficient fine-tuning. *arXiv preprint arXiv:2405.12130*.

RE Kass, L Tierney, and JB Kadane. 1990. The validity of posterior expansions based on laplace's method." essays in honor of george barnard, eds. s. geisser, js hodges.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

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. In *Forty-first International Conference on Machine Learning*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

David JC MacKay. 1992. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472.

Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716.

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. *Advances in Neural Information Processing Systems*, 37:121038–121072.

Mengjie Qian, Siyuan Tang, Rao Ma, Kate M Knill, and Mark JF Gales. 2024. Learn and don't forget: Adding a new language to asr foundation models. *arXiv preprint arXiv:2407.06800*.

P Rajpurkar. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Weijieying Ren, Xinlong Li, Lei Wang, Tianxiang Zhao, and Wei Qin. 2024. Analyzing and reducing catastrophic forgetting in parameter efficient tuning. *arXiv preprint arXiv:2402.18865*.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. 2018. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*.

Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. Online structured laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems*, 31.

Masaki Saito, Eiichi Matsumoto, and Shunta Saito. 2017. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE international conference on computer vision*, pages 2830–2839.

Alexandra Senderovich, Ekaterina Bulatova, Anton Obukhov, and Maxim Rakhuba. 2022. Towards practical control of singular values of convolutional layers. *Advances in Neural Information Processing Systems*, 35:10918–10930.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti

Bhosale, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018a. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. 2025. Milora: Harnessing minor singular components for parameter-efficient llm finetuning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4823–4836.

Liyuan Wang, Mingtian Zhang, Zhongfan Jia, Qian Li, Chenglong Bao, Kaisheng Ma, Jun Zhu, and Yi Zhong. 2021. Afec: Active forgetting of negative transfer in continual learning. *Advances in Neural Information Processing Systems*, 34:22379–22391.

Shaowen Wang, Linxi Yu, and Jian Li. 2024. Lora-ga: Low-rank adaptation with gradient approximation. *Advances in Neural Information Processing Systems*, 37:54905–54931.

Wei Wang, Ming Yan, and Chen Wu. 2018b. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. *arXiv preprint arXiv:1811.11934*.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. 2023. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671.

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

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Shipeng Yan, Jiangwei Xie, and Xuming He. 2021. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3014–3023.

Shu Yang, Muhammad Asif Ali, Cheng-Long Wang, Lijie Hu, and Di Wang. 2024a. Moral: Moe augmented lora for llms' lifelong learning. *arXiv preprint arXiv:2402.11260*.

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Leon Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. 2024b. Corda: Context-oriented decomposition adaptation of large language models. *arXiv preprint arXiv:2406.05223*.

Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. 2023. Neural collapse inspired feature-classifier alignment for few-shot class incremental learning. *arXiv preprint arXiv:2302.03004*.

Michael S Yao and Michael S Hansen. 2022. A path towards clinical adaptation of accelerated mri. *Proceedings of machine learning research*, 193:489.

Hao Yu and Jianxin Wu. 2023. Compressing transformers: features are low-rank, but weights are not! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11007–11015.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M Susskind. 2023. Stabilizing transformer training by preventing attention entropy collapse. In *International Conference on Machine Learning*, pages 40770–40803. PMLR.

Jia-Chen Zhang, Yu-Jie Xiong, He-Xi Qiu, Dong-Hai Zhu, and Chun-Ming Xia. 2024. Lora$^2$: Multi-scale low-rank approximations for fine-tuning large language models. *arXiv preprint arXiv:2408.06854*.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

## A Reproducibility Statement

In an effort to ensure reproducibility, we report the description of dataset in Section M.1.1, Section M.2.1 and Section M.3.1. Also we report the best hyperparameters of our experiments in Section M.1.3, Section M.2.3 and Section M.3.3. Our **SCLoRA** code to reproduce the experiment can be found at `https://bit.ly/3ElHoYb`.

## B Ethical Consideration

We utilized publicly available datasets, including GLUE, SQuAD and commonsense reasoning, which are commonly employed in academic research, and all sources have been appropriately cited. This research does not involve any personal or confidential information, thereby eliminating concerns related to privacy. Our proposed approach and the resulting insights contribute to the advancement of artificial intelligence while adhering to principles of ethical innovation and responsibility.

## C Broader Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted.

## D Use of LLMs

In accordance with ACL 2026 policy, we acknowledge the use of LLMs in the preparation of this paper. Their use was limited solely to improving translation accuracy and ensuring grammatical correctness.

## E Formulation & Additional Visualization of Fisher Overlap

Following Kirkpatrick et al. (2017), to examine whether different tasks solved by the same network rely on overlapping parameter subsets (see Fig. 1 (a)), we assessed the similarity of each task's Fisher information matrix. Specifically, we first computed the Fisher matrices for the two tasks, denoted by $F_1$ and $F_2$. We then normalized each matrix so that its trace was equal to 1, yielding $\hat{F}_1$ and $\hat{F}_2$. Next, we measure how closely these matrices aligned by computing the Fréchet distance, a metric on positive-semidefinite matrices, given as:

$$d^2(\hat{F}_1, \hat{F}_2) = \frac{1}{2}\mathrm{tr}\big(\hat{F}_1 + \hat{F}_2 - 2(\hat{F}_1\hat{F}_2)^{1/2}\big) \quad (11)$$

$$= \frac{1}{2}\|\hat{F}_1^{1/2} - \hat{F}_2^{1/2}\|_F, \quad (12)$$

where this quantity lies between 0 and 1. We then define the *overlap* of the two tasks' Fisher matrices as $1 - d^2$. Hence, an overlap of 0 implies that the two tasks employ entirely distinct sets of parameters, whereas an overlap of 1 indicates that one Fisher matrix is simply a scaled version of the other (i.e., $F_1 = \alpha F_2$ for some $\alpha > 0$).

Then, to verify the Fisher overlap across the singular components of the pre-trained network parameters, we perform SVD on the pre-trained parameters of the $\text{RoBERTa}_{\text{base}}$ model, trained on multiple datasets including BookCorpus. We group the singular components sorted by their singular values and reconstruct partial versions of the parameters from each group. Fig. 4 shows the Fisher overlap computed for the BookCorpus task as well as for each task in the GLUE benchmark. Across all tasks, the Fisher overlap progressively decreases as we move from groups containing larger singular values to those with smaller ones, indicating that fine-tuning increasingly struggles to reuse the finer-grained partial reconstructions of the pre-trained parameters.

## F Proof of Theorem 3.1

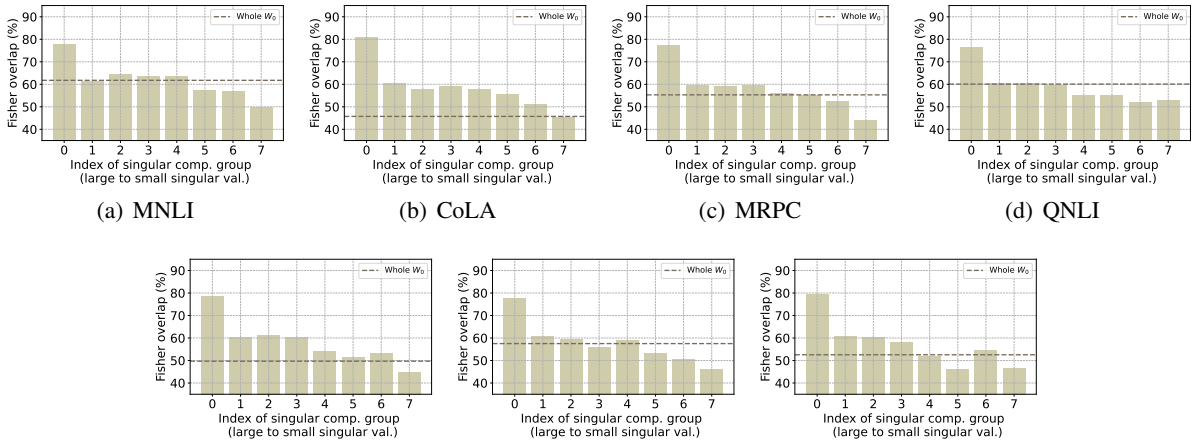
*Proof.* The optimization of neural networks can be considered as a process of estimating the network parameters $\theta$ through maximum a posteriori (MAP) estimation using the training data. This involves the pre-training dataset $\mathcal{D}_A$ and the fine-tuning dataset $\mathcal{D}_B$. The pre-trained weights are denoted as $\theta_0$, and the fine-tuned weights are represented as $\theta$.

The posterior to be maximized in the MAP estimation is formulated as:

$$p(\theta|\mathcal{D}_A, \mathcal{D}_B) = \frac{p(\mathcal{D}_B|\theta, \mathcal{D}_A)p(\theta|\mathcal{D}_A)}{p(\mathcal{D}_B|\mathcal{D}_A)} = \frac{p(\mathcal{D}_B|\theta)p(\theta|\mathcal{D}_A)}{p(\mathcal{D}_B|\mathcal{D}_A)}. \quad (13)$$

Taking a logarithm of the posterior, the MAP objective becomes:

$$\theta^* = \underset{\theta}{\mathrm{argmax}} \log p(\theta|\mathcal{D}_A, \mathcal{D}_B) = \underset{\theta}{\mathrm{argmax}} \big[\log p(\mathcal{D}_B|\theta) + \log p(\theta|\mathcal{D}_A)\big]. \quad (14)$$




(a) MNLI (b) CoLA (c) MRPC (d) QNLI

(e) QQP (f) RTE (g) SST-2

Figure 4: The Fisher overlap (Kirkpatrick et al., 2017) between the pre-trained task (BookCorpus) and the fine-tuning task (MRPC), evaluated over partial reconstructions of the pre-trained network parameters obtained by grouping singular components sorted from large to small according to their singular values.

Since the true posterior is intractable, we approximate the posterior using Laplace Approximation. $\log p(\theta|\mathcal{D}_A)$ can be expressed as a function $f(\theta)$ and approximated near the optimal point $f(\theta_0)$, where $\theta_0$ represents the pre-trained parameters, and $\nabla f(\theta_0) = 0$. Subsequently, a second-order Taylor expansion of $f(\theta)$ around $\theta_0$ is performed as follows:

$$\begin{aligned}\log p(\theta|\mathcal{D}_A) &\simeq f(\theta_0) + \frac{1}{2}(\theta-\theta_0)\nabla^2 f(\theta_0)(\theta-\theta_0) \\ &= f(\theta_0) + \frac{1}{2}(\theta-\theta_0)^\top H(\theta-\theta_0),\end{aligned} \tag{15}$$

where $H$ denotes the Hessian matrix of $f(\theta)$ evaluated at $\theta_0$. The expected value of the Hessian over the data distribution corresponds to the Fisher information matrix (FIM) $F$, defined as $F = -\mathbb{E}_{\mathcal{D}_A}[H]$. Following (MacKay, 1992; Kirkpatrick et al., 2017), we approximate the posterior as a Gaussian distribution with mean given by the parameters $\theta_0$ and a diagonal precision given by the diagonal of the Fisher information matrix $F$. Given this approximation, the log probability can be expressed as:

$$\begin{aligned}\log p(\theta|\mathcal{D}_A) &\simeq \log \hat{p}(\theta|\mathcal{D}_A) \\ &= f(\theta_0) - \frac{1}{2}(\theta-\theta_0)^\top F(\theta-\theta_0),\end{aligned} \tag{16}$$

where the $F$ is symmetric and positive semi-definite, i.e., for any vector $v$, $vFv^\top \geq 0$. Then using the singular value decomposition (SVD) on $\theta - \theta_0 = \Delta\theta = U\Sigma V^\top$ where $U \in \mathbb{R}^{d_1 \times r}$, $V \in \mathbb{R}^{r \times d_2}$, and $\Sigma \in \mathbb{R}^r$ with $\{\sigma_n\}_{1 \leq n \leq r}$. As $U, V$ are orthonormal singular vectors and $F$ is a positive semi-definite matrix,

$$\Delta\theta^\top F \Delta\theta \geq \lambda_{\min}(F)\|\Delta\theta\|_F = \lambda_{\min}(F)\|\Sigma\|_F. \tag{17}$$

Therefore, the log probability of the approximated prior for the fine-tuning task from the pre-trained task is upper bounded as:

$$\begin{aligned}\log \hat{p}(\theta|\mathcal{D}_A) &= f(\theta_0) - \tfrac{1}{2}(\theta-\theta_0)^\intercal F(\theta-\theta_0) \\ &\leq f(\theta_0) - \lambda_{\min}(F)\,\|\Sigma\|_F \\ &= f(\theta_0) - \lambda_{\min}(F)\sqrt{\sum_{n=1}^{r}\sigma_n^2}.\end{aligned} \tag{18}$$

□

## G Well-Established Properties on Laplace Approximation

### G.1 Error bound of Laplace Approximation

In Bayesian inference, one of the most widely used methods for approximating the posterior distribution is the Laplace approximation (Kass et al., 1990; Kirkpatrick et al., 2017; Ritter et al., 2018; Wang et al., 2021; Matena and Raffel, 2022; Gawlikowski et al., 2023). This method expands the log-posterior function around its mode (i.e., the MAP estimate) using a Taylor series and retains terms up to the second order, thereby approximating the posterior distribution by a Gaussian distribution. In other

words, higher-order terms beyond the quadratic expansion are discarded, and the resulting approximation error is generally limited and asymptotically negligible.

In particular, under standard regularity conditions, it is well established that the relative error of Laplace approximation is no worse than $\mathcal{O}_p(n^{-1})$ under standard regularity conditions, where $\mathcal{O}_p$ refers to stochastic boundedness (Kass et al., 1990; Bilodeau et al., 2023). This ensures that, as the sample size $n$ increases, the approximation error vanishes at the rate of $n^{-1}$. Moreover, in deep learning, where the number of training samples $n$ is typically very large, the accuracy of the Laplace approximation is further reinforced. Consequently, the Laplace approximation provides not only a practical tool but also a theoretically justified method for posterior approximation in both Bayesian inference and large-scale probabilistic modeling.

### G.2 Decay Rate of the Integral over the Mode-Distant Region

Laplace approximation is applied when the target function is sharply concentrated around a single mode $\theta_0$ and decays rapidly as $\theta$ moves away from it. The method rewrites the integral in exponential form and then approximates the log-posterior by a second-order Taylor expansion around its mode. The region where $\theta - \theta_0$ is large corresponds to the tail of the function. The approximation error in this region should not be judged solely by the magnitude of $|\theta - \theta_0|$; rather, its actual contribution to the integral must be considered. Kass et al. (1990) rigorously demonstrates that this tail contribution is negligible. First, as the sample size increases, the likelihood function becomes increasingly peaked, so the posterior concentrates around the mode $\theta_0$. Second, the integral over regions distant from $\theta_0$ decays exponentially with $n$, that is, it is bounded by $\exp(-nc)$ for $c > 0$. Consequently, the integral over the mode-distant region converges to zero, and the principal contribution to the integral arises near the mode.

## H Exponential Decay of Singular Values

To find the best possible $n$-dimensional subspace $V_n$ such that the closest approximation $v \in V_n$ to $W$ minimizes the error $\|W - v\|_X$, the definition of Kolmogorov $n$-width is formulated as follows:

$$d_n(W, X) = \inf_{\substack{V_n \subset X \\ \dim V_n = n}} \inf_{v \in V_n} \|W - v\|_X, \quad (19)$$

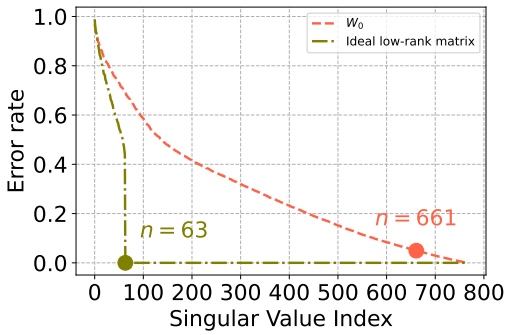



Figure 5: The error rate of the normalized singular values for: (i) the final output projection layer weights $W_0$ in the self-attention mechanism of DeBERTaV3$_{\text{base}}$, and (ii) an ideal low-rank matrix with rank $r = 64$. The marker indicates the $n$-value where the approximation error reaches 5%.

where $V_n$ is $n$-dimensional subspace of $X$, $v$ is an element from the subspace $V_n$. 'inf' stands for infimum. When using the Frobenius norm (or spectral norm) with matrices, the Kolmogorov $n$-width is computed by the singular values of $W$ as follows:

$$d_n(W, X) = \sigma_{n+1}, \quad (20)$$

where $\sigma_{n+1}$ is the $(n+1)$-th largest singular value of the matrix $W$. The Kolmogorov $n$-width measures how well a set $W$ can be approximated by an $n$-dimensional subspace. In other words, it represents the minimal maximum error when approximating with an $n$-dimensional subspace. Then we can determine the optimal dimensionality needed to achieve a desired approximation accuracy.

If the singular values decrease rapidly, $W$ can be well approximated even for small $n$, and the Kolmogorov $n$-width also decreases quickly. Therefore, the singular value decay rate $\alpha$, which plays a pivotal role in determining how effectively a matrix can be approximated, is commonly modeled by an exponential decay function as follows:

$$\sigma'_n = Ce^{-\alpha n}, \quad (21)$$

where $\sigma'_n$ represents the $n$-th modeled singular values, $C > 0$ is a constant, and $\alpha > 0$ is the decay rate. When the decay rate $\alpha$ is low, the singular values decrease gradually, resulting in large errors when approximating with the same $n$ dimensions. To minimize the approximation errors, a larger $n$ is required, indicating that significant information is contained in the lower singular values.

**Empirical analysis of the Kolmogorov $n$-width.** To empirically analyze the Kolmogorov $n$-width of the pre-trained language model, we present error rates based on low-rank approximation under the same conditions as shown in Fig. 1 (b) of Introduction. The formulation of error rates $E_W(n)$ is as follows:

$$E_W(n) = \left( \frac{\|W - v\|_F}{\|W\|_F} \right) \times 100\%, \qquad (22)$$

where $W$ is the original matrix and $v$ is the approximated matrix obtained by truncating the SVD to rank $n$. The error rates for the pre-trained model and the ideal low-rank matrix are presented in Fig. 5, with markers indicating the $n$-value where the error rate reaches 5%. For the ideal low-rank matrix, the rank at which the error rate reaches 5% is 63. This suggests that the matrix has a low-dimensional structure, with the most important information concentrated in the top singular values. The lower singular values have little effect on the approximation and can be considered noise. In contrast, for the pre-trained model, the $n$-value required to reach 95% approximation is 661, which is significantly larger than ideal low rank matrix. This indicates that the data is complex and high-dimensional, and the lower singular values contain important information rather than merely noise.

## I Discussion on Constraining the Frobenius Norm of $\Delta W$

Theorem 3.1 provides an upper bound of approximated posterior based on the Frobenius norm of $\Delta\theta$ and the minimum eigenvalue of the Fisher Information Matrix (FIM), $\lambda_{\min}(F)$. However, simply constraining the Frobenius norm is not sufficient. This is because the FIM $F$ encodes information about *important directions* about the pre-trained task in the parameter space. Specifically, Equation (8) provides the approximated log posterior of pre-trained task using Laplace approximation. The FIM $F$ is a positive semi-definite matrix. The quadratic term quantifies how much the parameter shift $\Delta\theta = \theta - \theta_0$ affects the output distribution of the original task. In low-rank update methods like LoRA, the parameter update can be expressed as $\Delta\theta = U\Sigma V^\top$. Due to the low-rank nature of the update, the singular values $\sigma_i$ often become concentrated in a small number of singular directions.

Even for the same Frobenius norm of $\Delta\theta$, if $\sigma_n$ is heavily concentrated in certain directions, and those directions align with sensitive ones in the FIM (i.e., those with large eigenvalues), the penalty term $\Delta\theta^\top F \Delta\theta$ can become significantly larger (not $\Delta\theta^\top \Delta\theta$). As a result, the approximated posterior decreases more sharply, which intensifies catastrophic forgetting on the pre-trained task. Therefore, the Frobenius norm merely controls the total magnitude of change, but does not prevent the change from being concentrated in directions that are crucial for the original task. In contrast, **SCLoRA** explicitly controls individual singular values via clipping, thereby directly limiting updates in directions where the model is most sensitive. This goes beyond a simple norm constraint and introduces a structural bias that suppresses updates in forgetting-prone directions. Therefore, while the Frobenius norm constrains only the total size of the update, **SCLoRA** enables spectral control via clipping over which components grow, effectively mitigating catastrophic forgetting in a more targeted and principled way. This is also well-described in Kirkpatrick et al. (2017), where L2-norm cannot mitigate catastrophic forgetting because important parameters from previous tasks are not adequately preserved. The standard L2-norm regularization treats all parameters uniformly, failing to account for their varying importance, and thus cannot effectively mitigate this problem.

## J Algorithm of SCLoRA

In this section, we summarize the detailed algorithm of **SCLoRA** in Algorithm 1.

**Algorithm 1** How to train **SCLoRA**

**Input:** Dataset $\mathcal{D}$; total iterations $T$; learning rate $\eta$; $\gamma$, $\bar{\sigma}_k$.

**for** $t = 1, \ldots, T$ **do**

$\Sigma_k^{(t)} = \min(\max(\Sigma_k^{(t)}, 0), \bar{\sigma}_k)$

$W_k^{(t)} = W_0 + U_k^{(t)} \Sigma_k^{(t)} (V_k^{(t)})^\top$

$U_k^{(t+1)} = U_k^{(t)} - \eta \nabla_{U_k} (\mathcal{L}(U_k^{(t)}, \Sigma_k^{(t)}, V_k^{(t)}) + \gamma R(U_k^{(t)}, V_k^{(t)}))$

$V_k^{(t+1)} = V_k^{(t)} - \eta \nabla_{V_k} (\mathcal{L}(U_k^{(t)}, \Sigma_k^{(t)}, V_k^{(t)}) + \gamma R(U_k^{(t)}, V_k^{(t)}))$

$\Sigma_k^{(t+1)} = \Sigma_k^{(t)} - \eta \nabla_{\Sigma_k} \mathcal{L}(U_k^{(t)}, \Sigma_k^{(t)}, V_k^{(t)})$

**end**

**Output:** The fine-tuned parameters $\{U^{(T)}, \Sigma^{(T)}, V^{(T)}\}$; $W^{(T)} = W_0 + U^{(T)} \Sigma^{(T)} (V^{(T)})^\top$.

## K LoRA with Discrete Fourier Transform

Discrete Fourier Transform (DFT) (Briggs and Henson, 1995) converts a finite sequence of equally-spaced samples of a function into a same-length sequence of equally-spaced samples of the discrete-time Fourier transform (DTFT), which is a complex-valued function of frequency. Recent studies have proposed the DFT-based parameter efficient fine-tuning method. FouRA (Borse et al., 2024) transforms the hidden vectors in the latent space into the singular domain using DFT and compute LoRA, followed by reconstruction through inverse DFT. FourierFT (Gao et al., 2024) considers the adapter as a 2D spatial-domain matrix and transforms into a 2D DFT spectrum. Therefore, existing methods have focused on DFT-based approaches to either flexibly select adapter ranks depending on the input or reduce the number of parameters. On the other hand, **SCLoRA** leverages a parameterized SVD in the parameter space, and injects the singular components with spectral clipping to achieve effective adaptation and mitigation of catastrophic forgetting.

We conduct additional experiments following experimental setup of FourierFT (Gao et al., 2024), one of the LoRA variants that employs the concept of DFT. Strictly following (Gao et al., 2024), we fine-tune $\text{RoBERTa}_{\text{base}}$, adapting only the query and value projection matrices using LoRA. We maintain the same experimental settings as the original work, while only searching for the learning rate from $\{1 \times 10^{-3}, 8 \times 10^{-4}, 6 \times 10^{-4}, \}$, $\gamma$ from $\{1 \times 10^{-2}, 3 \times 10^{-3}, 1 \times 10^{-3}\}$, and $q$ from $\{0.25, 0.5, 0.75, 1.0\}$ and the results are reported in Section K.

## L How does the Adapters $\Delta W$ Compared to $W$?

We explore the relationship between $\Delta W$ and $W$ by measuring the correlation between $\Delta W$ and $W$ as well as the magnitude of $\Delta W$ in comparison to its corresponding directions in $W$. To do so, we introduce two key factors:

- $\texttt{Factor}_{W \rightarrow \Delta W}$ is a factor formulated as $\|\Delta W\|_F / \|U_{\Delta W}^\top W V_{\Delta W}\|_F$, which indicates the ratio of the norm of difference over the norm of projected $W$ on the $r$-dimensional subspace of $\Delta W$. This factor is also called *amplification factor* (Hu et al., 2021), measuring how the new information of $\Delta W$ is related to the existing information of $W$. A larger ratio refers that the task-specific information of $W$ has been amplified in $\Delta W$.

- $\texttt{Factor}_{\Delta W \rightarrow W}$ is a factor formulated as $\|\Delta W\|_F / \|U_W^\top \Delta W V_W\|_F$, which is the ratio of the norm of difference over the norm of projected $\Delta W$ on the $r$-dimensional subspace of $W$. It indicates the extent to which the change aligns with $W$. A larger ratio refers that $\Delta W$ has learned new information that is not present in $W$.

Following Hu et al. (2021), we project $W$ onto the $r$-dimensional subspace of $\Delta W$ by computing $U^\top W V$, where $U$ and $V$ are the left and right singular vectors of $\Delta W$, $W$, and the random matrix. Additionally, we project $\Delta W$ onto the subspace of $W$ by computing $U^\top \Delta W V$. As shown in Section L, **SCLoRA** and other methods exhibit similar Frobenius norms when $W$ is projected onto the subspace of $\Delta W$, $W$ and random matrix. However, compared to the baselines, the projection of $\Delta W$ onto the subspace of $W$ in **SCLoRA** shows the lowest correlation with a value of 0.02, which is less than half of the smallest baseline. This suggests that **SCLoRA** processes the existing information in $W$ similarly to other methods, while being better at learning independent new information without relying on the existing information in $W$. Furthermore, considering the Frobenius norm of $\Delta W$, both LoRA and PiSSA exhibit a large $\texttt{Factor}_{W \rightarrow \Delta W}$ and a small $\texttt{Factor}_{\Delta W \rightarrow W}$, indicating that $\Delta W$ primarily amplifies information already present in $W$. MiLoRA also shows a large $\texttt{Factor}_{\Delta W \rightarrow W}$, but this results from the large magnitude of $\Delta W$, leading to significant changes from the pre-trained weights. In contrast, **SCLoRA** exhibits a relatively small $\texttt{Factor}_{W \rightarrow \Delta W}$ of 4.25 but a large $\texttt{Factor}_{\Delta W \rightarrow W}$ of 46.77. Given the small magnitude of $\Delta W$, this indicates that **SCLoRA** stands out for its ability to learn new information that is not already in $W$ with minimal deviation from the pre-trained weights. We define the following ratio based on these two factors:

$$\texttt{Ratio} = \frac{\texttt{Factor}_{\Delta W \rightarrow W}}{\texttt{Factor}_{W \rightarrow \Delta W}}. \tag{23}$$

A higher Ratio indicates that $\Delta W$ contains new components that do not lie within the subspace of the pre-trained weights. This value is significantly higher than those of existing LoRA-based methods,

| Method | # Params | SST-2 | MRPC | CoLA | QNLI | RTE | STS-B | Avg. |
|---|---|---|---|---|---|---|---|---|
| FT | 125M | $94.8_{\pm 0.2}$ | $90.2_{\pm 0.6}$ | $63.6_{\pm 2.6}$ | $92.8_{\pm 0.2}$ | $78.7_{\pm 0.7}$ | $91.2_{\pm 0.5}$ | 85.2 |
| BitFit | 0.1M | $93.7_{\pm 0.3}$ | $\mathbf{92.7_{\pm 0.6}}$ | $62.0_{\pm 1.9}$ | $91.8_{\pm 0.2}$ | $\underline{81.5}_{\pm 0.8}$ | $90.8_{\pm 0.6}$ | $\underline{85.4}$ |
| Adpt$^D$ | 0.3M | $94.2_{\pm 0.1}$ | $88.5_{\pm 1.1}$ | $60.8_{\pm 1.1}$ | $93.1_{\pm 0.4}$ | $71.5_{\pm 2.7}$ | $89.7_{\pm 0.7}$ | 83.0 |
| Adpt$^D$ | 0.9M | $94.7_{\pm 0.3}$ | $88.4_{\pm 0.1}$ | $62.6_{\pm 0.9}$ | $93.0_{\pm 0.2}$ | $75.9_{\pm 0.4}$ | $90.3_{\pm 0.2}$ | 84.2 |
| LoRA | 0.3M | $\mathbf{95.1_{\pm 0.2}}$ | $89.7_{\pm 0.7}$ | $63.4_{\pm 1.0}$ | $\underline{93.3}_{\pm 0.2}$ | $78.4_{\pm 0.2}$ | $\mathbf{91.5_{\pm 0.2}}$ | 85.2 |
| AdaLoRA | 0.3M | $94.5_{\pm 0.2}$ | $88.7_{\pm 0.5}$ | $62.0_{\pm 0.5}$ | $93.1_{\pm 0.3}$ | $81.0_{\pm 0.9}$ | $90.5_{\pm 0.2}$ | 85.0 |
| DyLoRA | 0.3M | $94.3_{\pm 0.2}$ | $89.5_{\pm 0.6}$ | $61.1_{\pm 0.9}$ | $92.2_{\pm 0.4}$ | $78.7_{\pm 1.0}$ | $91.1_{\pm 0.3}$ | 84.5 |
| FourierFT | 0.024M | $94.2_{\pm 0.3}$ | $90.0_{\pm 0.3}$ | $\underline{63.8}_{\pm 1.6}$ | $92.2_{\pm 0.1}$ | $79.1_{\pm 0.2}$ | $90.8_{\pm 0.4}$ | 85.0 |
| **SCLoRA** | 0.3M | $\underline{95.0}_{\pm 0.2}$ | $\underline{90.6}_{\pm 1.1}$ | $\mathbf{65.1}_{\pm 0.1}$ | $\mathbf{93.4}_{\pm 0.2}$ | $\mathbf{83.0}_{\pm 1.0}$ | $\underline{91.3}_{\pm 0.1}$ | **86.4** |

Table 6: Comparison of various methods with RoBERTa$_{\text{base}}$ on GLUE tasks with the experimental setup from Gao et al. (2024). The results for the baselines are copied from Gao et al. (2024). The best performance is set in **bold**, and the second best is set in underline.

| Model | $\|U^\top WV\|_F$ | | | $\|U_{W_q}^\top \Delta W V_{W_q}\|_F$ | $\|\Delta W\|_F$ | $\texttt{Factor}_{W \to \Delta W}$ | $\texttt{Factor}_{\Delta W \to W}$ |
|---|---|---|---|---|---|---|---|
| | $\Delta W_q$ | $W_q$ | Random | | | | |
| LoRA | 0.48 | 11.22 | 0.32 | 0.16 | 3.81 | 7.94 | 23.82 |
| PiSSA | 0.38 | 11.22 | 0.35 | 0.11 | 2.49 | 6.54 | 22.60 |
| MiLoRA | 0.45 | 11.22 | 0.35 | 0.08 | 3.11 | 6.91 | 38.86 |
| **SCLoRA** | 0.36 | 11.22 | 0.38 | 0.03 | 0.94 | 2.60 | 31.18 |

Table 7: The Frobenius norm of $U^\top WV$, where $U$ and $V$ are the left and right top $r$ singular vector directions of either: (1) $\Delta W_q$, (2) $W_q$, or (3) a random matrix. (4) The Frobenius norm of $U^\top \Delta WV$, where $U$ and $V$ are from $W_q$. (5) The Frobenius norm of $\Delta W$. (6,7) The introduced factors. The weights are taken from the last query layer of RoBERTa$_{\text{base}}$, fine-tuned on STS-B dataset with $r = 8$.

demonstrating that **SCLoRA** is not merely amplifying existing directions, but instead learning new task-specific directions that do not exist in the pre-trained weights. According to Section L, **SCLoRA** achieves a $\texttt{Ratio}$ of $31.18/2.60 \simeq 11.99$, which is significantly higher than other LoRA-based methods. This provides strong evidence that **SCLoRA** is not merely amplifying existing directions, but rather learning new information that is not already in $W$ from the pre-trained weights. In summary, **SCLoRA** preserves pre-trained knowledge while injecting fine-grained information into low-alignment regions, thereby naturally inducing a divergence of subspaces during fine-tuning.

# M Experimental Setup

## M.1 Natural Language Understanding

### M.1.1 Dataset description

We describe the benchmark datasets of GLUE (Wang et al., 2018a) below [1].

- **CoLA.** The Corpus of Linguistic Acceptability (Warstadt et al., 2019) provides a dataset of English sentences, where each sentence is judged for grammatical acceptability based on data from books and journal articles. The objective is a binary classification to determine whether a sentence is grammatically correct or incorrect. The dataset consists of 8.5k samples for training, 1k samples for validation, and 1k samples for test.

- **SST-2.** The Stanford Sentiment Treebank (Socher et al., 2013) includes sentences from movie reviews, along with human-provided sentiment annotations. The goal is to classify the sentiment of each sentence as either positive or negative. The dataset consists of 67k samples for training, 872 samples for validation, and 1.8k samples for test.

- **MRPC.** The Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005) contains pairs of sentences automatically extracted from online news sources. Human annotators label each pair, and the task is to identify whether the two sentences in a pair convey the same meaning. The dataset consists of 3.7k samples for training, 408 samples for validation, and 1.7k samples for test.

[1] https://huggingface.co/datasets/nyu-mll/glue

- **QQP.** The Quora Question Pairs dataset (Chen et al., 2018) consists of question pairs taken from Quora, a community-driven question-and-answer platform. The task is to determine if two given questions are semantically identical. The dataset consists of 364k samples for training, 40k samples for validation, and 391k samples for test.

- **MNLI.** The Multi-Genre Natural Language Inference Corpus (Williams et al., 2017) includes sentence pairs with textual entailment annotations collected through crowdsourcing. Given a premise and a hypothesis, the task is to predict whether the premise entails the hypothesis, contradicts it, or is neutral. The dataset includes both in-domain and cross-domain evaluations using a hidden test set. The dataset consists of 393k samples for training, 20k samples for validation, and 20k samples for test.

- **QNLI.** The Question-Answering Natural Language Inference dataset (Wang et al., 2018b) consists of question-paragraph pairs from which an answer must be found. The task involves determining whether a specific sentence from the paragraph answers the corresponding question. The dataset consists of 108k samples for training, 5.7k samples for validation, and 5.7k samples for test.

- **RTE.** The Recognizing Textual Entailment dataset (Bentivogli et al., 2009) comes from a series of annual challenges focusing on textual entailment. The task is to classify sentence pairs as either entailment or non-entailment. The dataset consists of 2.5k samples for training, 276 samples for validation, and 3k samples for test.

- **STS-B.** The Semantic Textual Similarity Benchmark (Cer et al., 2017) features sentence pairs drawn from various sources, including news headlines and image captions, with human-assigned similarity scores. The task is a regression problem where the model must predict a similarity score ranging from 0 to 5. The dataset consists of 7k samples for training, 1.5k samples for validation, and 1.4k samples for test.

#### M.1.2 Experimental setup

We evaluate **SCLoRA** on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018a), which includes 3 categories of natural language understanding tasks: i) single-sentence (CoLA and SST-2); ii) similarity and paraphrasing (MRPC, QQP, and STS-B); iii) natural language inference tasks (MNLI, QNLI, and RTE). For a fair comparison, following Hu et al. (2021), we adopt the pre-trained RoBERTa$_{\text{base}}$ as the backbone model. We use 1 GPU of NVIDIA RTX A6000 for experiments. We report Matthews correlation for CoLA, Spearman correlations for STS-B, and accuracy scores for the other tasks. We conduct the experiments in Huggingface Framework [2].

#### M.1.3 Hyperparameters

To tune **SCLoRA** with RoBERTa$_{\text{base}}$, we search for the learning rate from $\{4\times10^{-4}, 5\times10^{-4}\}$, $q$ from $\{0.25, 0.5, 0.75, 1.0\}$ and $\gamma$ from $\{1\times10^{-1}, 7\times10^{-2}, 5\times10^{-2}, 3\times10^{-2}, 1\times10^{-2}, 1\times10^{-3}\}$. The learnable singular vectors $U/V$ can be initialized as i) random $r$ singular vectors of $W_0$, ii) $U$ with zeros, $V$ with random Gaussian initialization. To tune **SCLoRA** with DeBERTaV3$_{\text{base}}$, we search $q$ from $\{0.25, 0.5, 0.75, 1.0\}$ and $\gamma$ from $\{1\times10^{-1}, 1.1\times10^{-1}, 1\times10^{-2}, 5\times10^{-1}, 6\times10^{-1}\}$. The learnable singular vectors $U/V$ are be initialized as $U$ with zeros and $V$ with random Gaussian initialization. We report the best hyperparameters of **SCLoRA** in Section M.1.3 to Section M.1.3.

#### M.1.4 Experimental Result with standard deviations

We report the experimental results on GLUE tasks with standard deviation of Roberta$_{\text{base}}$ and DeBERTa$_{\text{base}}$V3 in Section M.1.4 and Section M.1.4, respectively. Note that the results for Section M.1.4 are copied from Zhang et al. (2023), the results with standard deviation are reported only for **SCLoRA**.

### M.2 Question Answering

#### M.2.1 Dataset description

We describe the benchmark dataset of SQuAD (Rajpurkar, 2016; Rajpurkar et al., 2018). The Stanford Question Answering Dataset (SQuAD) is a benchmark for reading comprehension, featuring questions based on Wikipedia articles. Each question

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

| Dataset | Learning rate | Batch size | #Epochs | Metric | $q$ | $\gamma$ | How to initialize $U, V$ |
|---|---|---|---|---|---|---|---|
| CoLA | $4 \times 10^{-4}$ | 32 | 25 | Matthews correlation | 0.5 | $3 \times 10^{-2}$ | random $r$ singular vectors |
| MNLI | $5 \times 10^{-4}$ | 32 | 7 | Accuracy | 0.5 | $1 \times 10^{-1}$ | 0, random Gaussian |
| MRPC | $4 \times 10^{-4}$ | 16 | 30 | Accuracy | 0.5 | $1 \times 10^{-2}$ | random $r$ singular vectors |
| QNLI | $4 \times 10^{-4}$ | 32 | 5 | Accuracy | 0.25 | $1 \times 10^{-1}$ | 0, random Gaussian |
| QQP | $5 \times 10^{-4}$ | 32 | 5 | Accuracy | 0.25 | $1 \times 10^{-3}$ | 0, random Gaussian |
| RTE | $5 \times 10^{-4}$ | 32 | 50 | Accuracy | 1.0 | $5 \times 10^{-2}$ | 0, random Gaussian |
| SST-2 | $5 \times 10^{-4}$ | 32 | 24 | Accuracy | 1.0 | $1 \times 10^{-1}$ | 0, random Gaussian |
| STS-B | $4 \times 10^{-4}$ | 32 | 25 | Pearson correlation | 0.25 | $1 \times 10^{-1}$ | 0, random Gaussian |

Table 8: Best hyperparameters for **SCLoRA** in natural language understanding for $\text{RoBERTa}_{\text{base}}$

| Dataset | Learning rate | Batch size | #Epochs | Metric | $q$ | $\gamma$ |
|---|---|---|---|---|---|---|
| CoLA | $8 \times 10^{-4}$ | 32 | 25 | Matthews correlation | 1.0 | $5 \times 10^{-1}$ |
| MNLI | $5 \times 10^{-4}$ | 32 | 7 | Accuracy | 0.75 | $1 \times 10^{-2}$ |
| MRPC | $1 \times 10^{-3}$ | 32 | 30 | Accuracy | 0.75 | $5 \times 10^{-1}$ |
| QNLI | $5 \times 10^{-4}$ | 32 | 5 | Accuracy | 0.75 | $1 \times 10^{-1}$ |
| QQP | $8 \times 10^{-4}$ | 32 | 5 | Accuracy | 0.25 | $1 \times 10^{-2}$ |
| RTE | $1.2 \times 10^{-3}$ | 32 | 50 | Accuracy | 0.75 | $1 \times 10^{-1}$ |
| SST-2 | $8 \times 10^{-4}$ | 32 | 24 | Accuracy | 1.0 | $1 \times 10^{-1}$ |
| STS-B | $2.2 \times 10^{-3}$ | 32 | 25 | Pearson correlation | 0.5 | $5 \times 10^{-1}$ |

Table 9: Best hyperparameters for **SCLoRA** in natural language understanding for $\text{DeBERTaV3}_{\text{base}}$ with $r = 8$

is answered with a specific text segment (or span) from the corresponding passage, though some questions may have no answer at all.

- **SQuADv1.1.** [3] Over 100,000 question-answer pairs derived from more than 500 articles. The dataset consists of 87,599 samples for training and 10,570 for validation.
- **SQuADv2.0.** [4] Combines the 100,000 questions in SQuADv1.1 with over 50,000 unanswerable questions to closely resemble answerable ones. To perform well on SQuADv2.0, systems must not only provide correct answers when available but also recognize when a question cannot be answered based on the given passage and abstain from responding. The dataset consists of 130,319 samples for training and 11,873 for validation.

#### M.2.2 Experimental setup

We evaluate **SCLoRA** on two question answering (QA) tasks: SQuAD v1.1 (Rajpurkar, 2016) and SQuADv2.0 (Rajpurkar et al., 2018). Following Zhang et al. (2023), we fine-tune a pre-trained $\text{DeBERTaV3}_{\text{base}}$ (He et al., 2021) with **SCLoRA** and set the rank $r$ of LoRA as $\{1, 2, 4, 8\}$. These tasks are considered as a sequence labeling problem, where the goal is to predict the probability of each token being the start and end of the answer span. We measured the performance of model using the Exact Match (EM) and F1 metrics. We use 1 GPU of NVIDA RTX 3090 24GB for experiments. We conduct the experiments in Huggingface Framework.

#### M.2.3 Hyperparameters

To tune **SCLoRA**, we fix the batch size as 16, and train 10 epochs for SQuADv1.1 and 12 epochs for SQuADv2.0, respectively. We search for the learning rate from $\{1 \times 10^{-3}, 2 \times 10^{-3}, 3 \times 10^{-3}\}$, $q$ from $\{0.5, 0.75, 1.0\}$ and $\gamma$ from $5 \times 10^{-1}, 1 \times 10^{-1}, 7 \times 10^{-2}, 5 \times 10^{-2}, 1 \times 10^{-2}\}$. The learnable singular vectors $U/V$ are be initialized as $U$ with zeros and $V$ with random Gaussian initialization. We report the best hyperparameters of **SCLoRA** in Section M.2.3.

### M.3 Commonsense Reasoning

#### M.3.1 Dataset description

The commonsense reasoning tasks are intended to require the model to go beyond pattern recognition. Instead, the model should use "common sense" or world knowledge to make inferences. The commonsense reasoning tasks comprise 8 sub-tasks, each with a predefined training and testing set.

- **BoolQ.** The model answers yes/no questions

[3] https://huggingface.co/datasets/rajpurkar/squad

[4] https://huggingface.co/datasets/rajpurkar/squad_v2

| Dataset | Learning rate | Batch size | #Epochs | Metric | $q$ | $\gamma$ |
|---|---|---|---|---|---|---|
| CoLA | $8 \times 10^{-4}$ | 32 | 25 | Matthews correlation | 1.0 | $1.1 \times 10^{-1}$ |
| MNLI | $5 \times 10^{-4}$ | 32 | 7 | Accuracy | 0.75 | $1 \times 10^{-1}$ |
| MRPC | $1 \times 10^{-3}$ | 32 | 30 | Accuracy | 0.5 | $1 \times 10^{-2}$ |
| QNLI | $7 \times 10^{-4}$ | 32 | 5 | Accuracy | 0.75 | $1 \times 10^{-1}$ |
| QQP | $8 \times 10^{-4}$ | 32 | 5 | Accuracy | 0.25 | $5 \times 10^{-3}$ |
| RTE | $1.2 \times 10^{-3}$ | 32 | 50 | Accuracy | 0.75 | $1 \times 10^{-1}$ |
| SST-2 | $8 \times 10^{-4}$ | 32 | 24 | Accuracy | 1.0 | $5 \times 10^{-1}$ |
| STS-B | $2.2 \times 10^{-3}$ | 32 | 25 | Pearson correlation | 1.0 | $6 \times 10^{-1}$ |

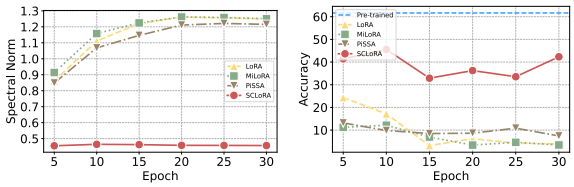
Table 10: Best hyperparameters for **SCLoRA** in natural language understanding for DeBERTaV3$_{\text{base}}$ with $r = 2$

| Method | MNLI | SST-2 | CoLA | QQP | QNLI | RTE | MRPC | STS-B | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| LoRA | $87.93_{\pm 0.15}$ | $94.80_{\pm 0.11}$ | $64.49_{\pm 0.64}$ | $90.94_{\pm 0.04}$ | $92.73_{\pm 0.11}$ | $80.39_{\pm 0.74}$ | $89.05_{\pm 0.12}$ | $90.87_{\pm 0.08}$ | 86.40 |
| AdaLoRA | $87.90_{\pm 0.09}$ | $94.80_{\pm 0.13}$ | $62.41_{\pm 0.66}$ | $90.16_{\pm 0.07}$ | $92.67_{\pm 0.12}$ | $81.59_{\pm 0.72}$ | $89.38_{\pm 0.57}$ | $91.08_{\pm 0.11}$ | 86.25 |
| PiSSA | $87.95_{\pm 0.01}$ | $94.53_{\pm 0.19}$ | $64.66_{\pm 0.98}$ | $90.97_{\pm 0.05}$ | $92.53_{\pm 0.14}$ | $79.18_{\pm 0.68}$ | $89.79_{\pm 1.41}$ | $90.96_{\pm 0.08}$ | 86.32 |
| MiLoRA | $87.88_{\pm 0.11}$ | $94.69_{\pm 0.30}$ | $64.31_{\pm 0.97}$ | $91.02_{\pm 0.04}$ | $92.96_{\pm 0.21}$ | $81.35_{\pm 1.23}$ | $89.30_{\pm 0.12}$ | $90.96_{\pm 0.05}$ | 86.56 |
| LoRA+ | $86.96_{\pm 0.65}$ | $93.92_{\pm 0.00}$ | $63.32_{\pm 0.69}$ | $90.69_{\pm 0.07}$ | $92.77_{\pm 0.01}$ | $81.59_{\pm 1.18}$ | $88.97_{\pm 0.35}$ | $90.84_{\pm 0.14}$ | 86.13 |
| LoRA-GA | $85.18_{\pm 0.16}$ | $93.16_{\pm 0.18}$ | $62.09_{\pm 0.80}$ | $88.57_{\pm 0.08}$ | $91.64_{\pm 0.13}$ | $76.77_{\pm 4.46}$ | $89.54_{\pm 0.62}$ | $90.87_{\pm 0.09}$ | 84.73 |
| CorDA | $86.28_{\pm 0.27}$ | $94.38_{\pm 0.53}$ | $62.43_{\pm 0.94}$ | $90.14_{\pm 0.11}$ | $92.36_{\pm 0.24}$ | $78.10_{\pm 0.55}$ | $89.87_{\pm 0.93}$ | $90.68_{\pm 0.14}$ | 85.53 |
| DoRA | $87.81_{\pm 0.04}$ | $95.11_{\pm 0.19}$ | $64.23_{\pm 0.10}$ | $90.65_{\pm 0.11}$ | $92.93_{\pm 0.10}$ | $81.35_{\pm 0.95}$ | $89.54_{\pm 0.23}$ | $91.01_{\pm 0.16}$ | 86.58 |
| **SCLoRA** | $87.95_{\pm 0.12}$ | $95.37_{\pm 0.29}$ | $64.79_{\pm 0.20}$ | $90.76_{\pm 0.08}$ | $93.09_{\pm 0.14}$ | $83.15_{\pm 0.17}$ | $90.32_{\pm 0.86}$ | $91.22_{\pm 0.05}$ | 87.08 |

Table 11: Comparison of various methods on GLUE tasks with different random seeds.

about short passages, testing its ability to understand statements.

- **PIQA.** The model chooses the most plausible solution for a physical interaction, focusing on practical reasoning.
- **SIQA.** The model infers the most suitable outcome or rationale in everyday social contexts.
- **HellaSwag.** The model selects the most coherent continuation of a short scenario, emphasizing commonsense inference.
- **WinoGrande.** The model resolves ambiguous pronoun references that require broad commonsense to disambiguate.
- **ARC-e.** The model tackles elementary-level science questions assessing basic scientific knowledge.
- **ARC-c.** The model addresses harder, more nuanced science questions requiring deeper reasoning.
- **OBQA.** OpenBookQA. The model answers questions using a provided 'open book' of facts, testing its ability to integrate and apply specific knowledge.




(a) Spectral Norm (b) Accuracy

Figure 6: Changes during fine-tuning RoBERTa$_{\text{base}}$ on the MRPC dataset: (a) Spectral norm of $\Delta W$ in the query layer; (b) accuracy on the pre-trained task (BookCorpus dataset).

### M.3.2 Experimental setup

Following Hu et al. (2023), we amalgamate the training datasets from all 8 tasks to create the final training dataset and conduct evaluations on the individual testing dataset for each task.

### M.3.3 Hyperparameters

To tune **SCLoRA**, we search for the learning rate from $\{3\times10^{-4}, 4\times10^{-4}\}$, $q$ from $\{0.5, 0.75, 1.0\}$ and $\gamma$ from $\{5 \times 10^{-1}, 1 \times 10^{-1}\}$. The learnable singular vectors $U/V$ are initialized as $U$ with zeros, $V$ with random Gaussian initialization. We report the best hyperparameters of **SCLoRA** in Section M.3.3.

| Method | # Params | MNLI | SST-2 | CoLA | QQP | QNLI | RTE | MRPC | STS-B | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| Full FT | 184M | 89.90 | 95.63 | 69.19 | 92.40 | 94.03 | 83.75 | 89.46 | 91.60 | 88.09 |
| BitFit | 0.10M | 89.37 | 94.84 | 66.96 | 88.41 | 92.24 | 78.70 | 87.75 | 91.35 | 86.02 |
| HAdapter | 1.22M | 90.13 | 95.53 | 68.64 | 91.91 | 94.11 | 84.48 | 89.95 | 91.48 | 88.12 |
| PAdapter | 1.18M | 90.33 | 95.61 | 68.77 | 92.04 | 94.29 | 85.20 | 89.46 | 91.54 | 88.24 |
| $\text{LoRA}_{\text{r=8}}$ | 1.33M | 90.65 | 94.95 | 69.82 | 91.99 | 93.87 | 85.20 | 89.95 | 91.60 | 88.34 |
| AdaLoRA | 1.27M | 90.76 | 96.10 | 71.45 | 92.23 | 94.55 | 88.09 | 90.69 | 91.84 | 89.31 |
| **SCLoRA** | 1.33M | $90.36_{\pm 0.03}$ | $96.33_{\pm 0.41}$ | $71.49_{\pm 0.60}$ | $92.33_{\pm 0.01}$ | $94.57_{\pm 0.05}$ | $89.41_{\pm 0.17}$ | $91.58_{\pm 1.21}$ | $92.19_{\pm 0.07}$ | 89.78 |
| HAdapter | 0.61M | 90.12 | 95.30 | 67.87 | 91.65 | 93.76 | 85.56 | 89.22 | 91.30 | 87.93 |
| PAdapter | 0.60M | 90.15 | 95.53 | 69.48 | 91.62 | 93.98 | 84.12 | 89.22 | 91.52 | 88.04 |
| HAdapter | 0.31M | 90.10 | 95.41 | 67.65 | 91.54 | 93.52 | 83.39 | 89.25 | 91.31 | 87.60 |
| PAdapter | 0.30M | 89.89 | 94.72 | 69.06 | 91.40 | 93.87 | 84.48 | 89.71 | 91.38 | 87.90 |
| $\text{LoRA}_{\text{r=2}}$ | 0.33M | 90.30 | 94.95 | 68.71 | 91.61 | 94.03 | 85.56 | 89.71 | 91.68 | 88.15 |
| AdaLoRA | 0.32M | 90.66 | 95.80 | 70.04 | 91.78 | 94.49 | 87.36 | 90.44 | 91.63 | 88.86 |
| **SCLoRA** | 0.33M | $90.66_{\pm 0.02}$ | $96.18_{\pm 0.14}$ | $71.83_{\pm 1.08}$ | $91.82_{\pm 0.07}$ | $94.50_{\pm 0.00}$ | $89.89_{\pm 1.47}$ | $91.83_{\pm 0.90}$ | $92.00_{\pm 0.23}$ | 89.83 |

Table 12: Performance comparison of various methods with $\text{DeBERTaV3}_{\text{base}}$ on GLUE tasks with different random seeds. The results for the baselines are copied from (Zhang et al., 2023).

| Dataset | $r$ | Learning rate | $q$ | $\gamma$ |
|---|---|---|---|---|
| SQuADv1.1 | 1 | $2 \times 10^{-3}$ | 0.5 | $5 \times 10^{-1}$ |
| | 2 | $2 \times 10^{-3}$ | 0.5 | $1 \times 10^{-1}$ |
| | 4 | $1 \times 10^{-3}$ | 0.5 | $1 \times 10^{-2}$ |
| | 8 | $2 \times 10^{-3}$ | 0.5 | $1 \times 10^{-1}$ |
| SQuADv2.0 | 1 | $2 \times 10^{-3}$ | 0.75 | $7 \times 10^{-2}$ |
| | 2 | $2 \times 10^{-3}$ | 0.75 | $5 \times 10^{-2}$ |
| | 4 | $3 \times 10^{-3}$ | 0.75 | $5 \times 10^{-1}$ |
| | 8 | $3 \times 10^{-3}$ | 0.75 | $5 \times 10^{-1}$ |

Table 13: Best hyperparameters for **SCLoRA** in question answering

| Model | Learning rate | Batch size | #Epochs | $q$ | $\gamma$ |
|---|---|---|---|---|---|
| LLaMA-7B | $3 \times 10^{-4}$ | 16 | 3 | 0.5 | 0.1 |
| LLaMA2-7B | $4 \times 10^{-4}$ | 16 | 3 | 0.75 | 0.5 |

Table 14: Best hyperparameters for **SCLoRA** in commonsense reasoning

## N Spectral analysis of $\Delta W$

Theorem 3.2 demonstrates that the spectral norm of large weight matrices increases rapidly when adaptive optimizers are applied. However, **SCLoRA** prevents this increase by restricting the range of the singular value of $\Delta W$ during fine-tuning, ensuring that the spectral norm does not grow excessively. To empirically verify this difference, Fig. 6 (a) illustrates the evolution of the spectral norm of $\Delta W$ across various methods during the fine-tuning with MRPC data on $\text{RoBERTa}_{\text{base}}$. While LoRA and its variants tend to increase the spectral norm during fine-tuning, **SCLoRA** maintains a smaller spectral norm. This suggests that, unlike other LoRA-based methods, **SCLoRA** adapts to fine-tuning tasks by effectively incorporating new information with the singular components with spectral clipping. Furthermore, consistent with Theorem 3.1, we show that the uncontrolled growth of adapter singular values induces catastrophic forgetting, which is empirically confirmed in Fig. 6(b). As fine-tuning progresses, the performance of baseline methods degrades sharply, whereas **SCLoRA** preserves approximately three-times higher performance, demonstrating its strong robustness against catastrophic forgetting.

## O Further Experiments on Mitigating Catastrophic Forgetting

In this section, we validate **SCLoRA** on mitigating the catastrophic forgetting, followed by Section 6.

### O.1 Mitigating catastrophic forgetting with other models

To further validate the effectiveness of **SCLoRA** in mitigating catastrophic forgetting, we report the performance of $\text{RoBERTa}_{\text{base}}$ models trained on SST-2, CoLA, RTE, and MRPC, along with the accuracy on their corresponding pre-training tasks using BookCorpus (Zhu et al., 2015), OpenWebText (Gokaslan et al., 2019), and STORIES (Trinh and Le, 2018) datasets in Table 15. Although the degree of catastrophic forgetting varies across existing LoRA models depending on the type of data, **SCLoRA** consistently adapts to new downstream tasks with improved fine-tuning performance while substantially preserving accuracy on the pre-training tasks. In particular, across all datasets, **SCLoRA** maintains much higher pre-training accuracy than LoRA while still achieving

| Dataset | Method | SST-2 | | CoLA | | RTE | | QNLI | | MRPC | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\text{Acc.}_{\text{ft}}$ | $\text{Acc.}_{\text{pt}}$ | $\text{Acc.}_{\text{ft}}$ | $\text{Acc.}_{\text{pt}}$ | $\text{Acc.}_{\text{ft}}$ | $\text{Acc.}_{\text{pt}}$ | $\text{Acc.}_{\text{ft}}$ | $\text{Acc.}_{\text{pt}}$ | $\text{Acc.}_{\text{ft}}$ | $\text{Acc.}_{\text{pt}}$ |
| BookCorpus | Pre-trained | - | 61.64 | - | 61.64 | - | 61.64 | - | 61.64 | - | 61.64 |
| | LoRA | 94.80 | 32.35 | 64.49 | 50.85 | 92.73 | 56.76 | 80.39 | 33.78 | 89.05 | 3.77 |
| | PiSSA | 94.53 | 27.14 | 64.66 | 44.93 | 92.53 | 52.75 | 79.18 | 10.85 | 89.79 | 4.78 |
| | MiLoRA | 94.69 | 27.24 | 64.31 | 51.38 | 92.96 | 52.67 | 81.35 | 25.99 | 89.30 | 2.50 |
| | LoRA-GA | 93.16 | 0.11 | 62.09 | 18.47 | 91.64 | 0.26 | 76.77 | 2.94 | 89.54 | 0.03 |
| | CorDA | 94.38 | 29.38 | 62.43 | 29.57 | 92.36 | 56.12 | 78.10 | 33.21 | 89.87 | 14.26 |
| | **SCLoRA** | **95.37** | **51.29** | **64.79** | **54.78** | **93.09** | **57.15** | **83.15** | **50.07** | **90.32** | **32.00** |
| OpenWebText | Pre-trained | - | 68.21 | - | 68.21 | - | 68.21 | - | 68.21 | - | 68.21 |
| | LoRA | 94.80 | 54.33 | 64.49 | 63.67 | 92.73 | 62.88 | 80.39 | 53.47 | 89.05 | 18.36 |
| | PiSSA | 94.53 | 48.49 | 64.66 | 60.28 | 92.53 | 58.96 | 79.18 | 45.97 | 89.79 | 17.92 |
| | MiLoRA | 94.69 | 49.22 | 64.31 | 64.25 | 92.96 | 60.58 | 81.35 | 51.74 | 89.30 | 16.93 |
| | LoRA-GA | 93.16 | 0.27 | 62.09 | 26.78 | 91.64 | 0.95 | 76.77 | 7.81 | 89.54 | 0.04 |
| | CorDA | 94.38 | 50.36 | 62.43 | 59.56 | 92.36 | 63.37 | 78.10 | 54.07 | 89.87 | 40.10 |
| | **SCLoRA** | **95.37** | **65.67** | **64.79** | **64.31** | **93.09** | **63.59** | **83.15** | **62.69** | **90.32** | **47.27** |
| STORIES | Pre-trained | - | 69.49 | - | 69.49 | - | 69.49 | - | 69.49 | - | 69.49 |
| | LoRA | 94.80 | 40.13 | 64.49 | 61.87 | 92.73 | 62.89 | 80.39 | 45.90 | 89.05 | 8.42 |
| | PiSSA | 94.53 | 37.27 | 64.66 | 59.20 | 92.53 | 57.15 | 79.18 | 31.86 | 89.79 | 7.75 |
| | MiLoRA | 94.69 | 37.47 | 64.31 | 61.46 | 92.96 | 60.30 | 81.35 | 36.13 | 89.30 | 4.28 |
| | LoRA-GA | 93.16 | 0.14 | 62.09 | 24.80 | 91.64 | 0.36 | 76.77 | 4.75 | 89.54 | 0.04 |
| | CorDA | 94.38 | 34.17 | 62.43 | 52.48 | 92.36 | 63.41 | 78.10 | 53.56 | 89.87 | 23.46 |
| | **SCLoRA** | **95.37** | **60.38** | **64.79** | **64.45** | **93.09** | **63.91** | **83.15** | **60.31** | **90.32** | **41.12** |

Table 15: Comparison on catastrophic forgetting with RoBERTa as the backbone model. '$\text{Acc.}_{\text{ft}}$' and '$\text{Acc.}_{\text{pt}}$' denote the accuracies on the fine-tuning and pre-training tasks, respectively.

the best fine-tuned task performance. Although MiLoRA modifies the minor components with the aim of preserving pre-trained knowledge, it still suffers from catastrophic forgetting during fine-tuning due to uncontrolled spectral growth. In addition, both CorDA and LoRA-GA exhibit even more severe forgetting, as they involve uncontrolled spectral updates and initialize the adapters using a subset of calibration samples from the fine-tuning task. This initialization is biased toward the downstream distribution, making the model more prone to drifting away from the pre-trained knowledge. In particular, LoRA-GA is designed to approximate the full gradient of the pre-trained weights on the fine-tuning task, and prior studies have shown that full-gradient adaptation tends to induce even greater forgetting (Biderman et al., 2024), which explains its more pronounced degradation of pre-trained knowledge. These results demonstrate that **SCLoRA** effectively mitigates catastrophic forgetting without sacrificing downstream effectiveness.

## O.2 Catastrophic forgetting with continual Learning

While our primary analysis focuses on single-task fine-tuning scenarios, catastrophic forgetting is inherently a continual learning problem. To more rigorously evaluate the effectiveness of **SCLoRA** in achieving stable adaptation while mitigating catastrophic forgetting, we therefore conduct additional experiments in continual learning settings. Specifically, we consider both a GLUE-based sequential fine-tuning setup and a standardized continual learning benchmark proposed in prior work.

### O.2.1 Continual learning on GLUE task

In the GLUE-based continual learning experiments, we sequentially fine-tune $\Delta W$ across multiple downstream tasks. For each task, hyperparameters such as the number of training epochs and batch size follow the baseline and reported optimal settings to ensure a fair comparison. After completing fine-tuning on each task, we evaluate both the performance on the corresponding downstream task and the accuracy on the pretraining dataset, BookCorpus. Compared with LoRA and the SVD-based method MiLoRA, **SCLoRA** exhibits a substantially slower degradation of pretraining performance while consistently achieving stronger adaptation to downstream tasks throughout the sequence. These results indicate that **SCLoRA** effectively preserves pretrained knowledge even when learning multiple tasks sequentially, enabling more stable and reliable task adaptation.

### O.2.2 Continual learning with benchmark

In addition, to assess performance under a more standardized continual learning protocol, we fol-

| Method | SST-2 | | CoLA | | RTE | | MRPC | |
|---|---|---|---|---|---|---|---|---|
| | $\text{Acc.}_{\text{ft}}$ | $\text{Acc.}_{\text{pt}}$ | $\text{Acc.}_{\text{ft}}$ | $\text{Acc.}_{\text{pt}}$ | $\text{Acc.}_{\text{ft}}$ | $\text{Acc.}_{\text{pt}}$ | $\text{Acc.}_{\text{ft}}$ | $\text{Acc.}_{\text{pt}}$ |
| Pre-train | – | 61.64 | – | 61.64 | – | 61.64 | – | 61.64 |
| LoRA | 94.50 | 45.15 | 59.74 | 48.51 | 71.00 | 40.47 | 88.97 | 33.38 |
| MiLoRA | 94.76 | 48.64 | 59.77 | 52.64 | 71.00 | 47.33 | 88.24 | 29.77 |
| **SCLoRA** | **95.37** | **51.76** | **60.77** | **54.04** | **79.18** | **52.44** | **89.30** | **45.08** |

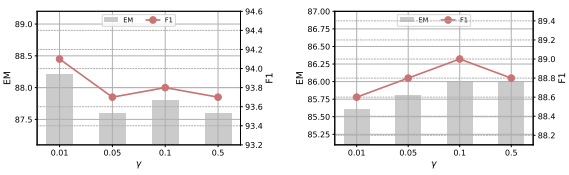
Table 16: Continual learning on GLUE. $\text{RoBERTa}_{\text{base}}$ are sequentially fine-tuned on SST-2 → CoLA → RTE → MRPC, and after each task we report both downstream accuracy and pre-training accuracy on BookCorpus.

| Method | Order-1 | Order-2 | Order-3 | Avg |
|---|---|---|---|---|
| SeqFT | 18.9 | 24.9 | 41.7 | 28.5 |
| SeqLoRA | 44.6 | 32.7 | 53.7 | 43.7 |
| IncLoRA | 66.0 | 64.9 | 68.3 | 66.4 |
| Replay | 55.2 | 56.9 | 61.3 | 57.8 |
| EWC | 48.7 | 47.7 | 54.5 | 50.3 |
| LwF | 54.4 | 53.1 | 49.6 | 52.3 |
| L2P | 60.3 | 61.7 | 61.1 | 60.7 |
| LFPT5 | 67.6 | 72.6 | **77.9** | 72.7 |
| O-LoRA | 75.4 | 75.7 | 76.3 | 75.8 |
| **SCLoRA** | **76.9**$_{\pm 1.0}$ | **76.8**$_{\pm 1.9}$ | 76.2$_{\pm 0.9}$ | **76.6** |

Table 17: Standard CL Benchmark

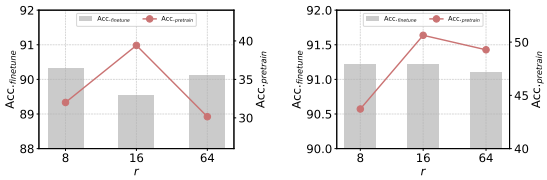
low the continual learning benchmark introduced in O-LoRA (Wang et al., 2023). We strictly follow the prescribed experimental protocol and evaluate three different task orders using the T5 model on five datasets — AG News, Amazon Reviews, Yelp Reviews, DBpedia, and Yahoo Answers — repeating each experiment three times. To ensure a fair comparison with O-LoRA in terms of parameter budget, we train a new adapter for each task while applying **SCLoRA** under the same parameter configuration. On average, **SCLoRA** outperforms existing methods in the continual learning setting on average. This further demonstrates that the spectral constraints imposed by **SCLoRA** effectively mitigate catastrophic forgetting not only in single-task fine-tuning but also in continual learning scenarios.

# P Additional Studies




(a) SQuADv1.1, $r$=4 (b) SQuADv2.0, $r$=4

Figure 7: Sensitivity studies on $\gamma$




(a) MRPC (b) STS-B

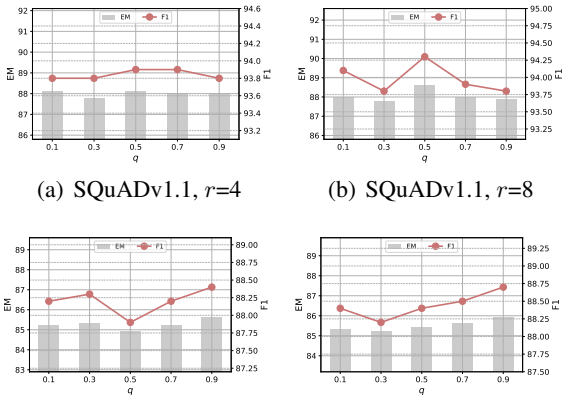
Figure 8: Sensitivity on $r$

## P.1 Sensitivity study on the orthogonal regularization coefficient $\gamma$

The orthogonal regularization applied to $U$ and $V$ is used to learn the singular values that consists the injected singular components. We conduct a sensitivity study on the orthogonal regularization coefficient $\gamma$ on Question-Answering (QA) tasks by fine-tuning the $\text{DeBERTaV3}_{\text{base}}$ model on the SQuAD v1.1/v2.0 datasets with rank $r = 4$. As shown in Fig. 7, the optimal value of $\gamma$ varies across datasets, indicating that different tasks prefer different regularization strengths. However, the performance does not degrade sharply outside the optimal range; instead, the model remains largely robust, exhibiting no severe sensitivity to the choice of $\gamma$.

## P.2 Sensitivity on the rank $r$

To evaluate how the choice of rank affects both downstream performance and catastrophic forgetting, we conduct a sensitivity study by varying the rank $r$. We measure performance using RoBERTa on two representative GLUE tasks—MRPC and STS-B—and assess the corresponding degradation on the BookCorpus pre-training dataset. The results are reported in Fig. 8. Although the optimal adapter rank and configuration vary across datasets and model sizes, and increasing the rank does not always yield consistent gains, we generally do not observe drastic performance degradation as the rank increases.



(a) SQuADv1.1, $r$=4 (b) SQuADv1.1, $r$=8

(c) SQuADv2.0, $r$=4 (d) SQuADv2.0, $r$=8

Figure 9: Sensitivity on $q$

### P.3 Additional sensitivity on $q$

Followed by Section 7.1, we conduct an additional sensitivity study on the $q$ on Question-Answering (QA) tasks. Fig. 9 illustrates that the fine-tuning performance does not change drastically as the quantile value varies. This indicates that using quantile-based bounds does not significantly degrade the performance of downstream tasks.

### P.4 Ablation on integrating SCLoRA with SVD-based LoRA

Existing SVD-based LoRA methods either apply SVD during the initialization stage of fine-tuning or use parameterized SVD to adjust the rank. However, these approaches generally do not explicitly control the spectral dynamics during the fine-tuning process. To examine whether SCLoRA can complement such methods by addressing this aspect, we conduct an ablation study in which our framework is integrated into existing baseline models. Specifically, for MiLoRA, we apply orthogonality regularization and singular value clipping after constructing the adapter using minor components, and for AdaLoRA, we incorporate singular value clipping within each allocated rank during the adaptive rank allocation procedure. All experiments use the best hyperparameters for both the backbone models and SCLoRA, and the results are reported in Table 18. Across downstream tasks, we observe consistent and modest performance improvements when combining SCLoRA with existing SVD-based methods (MiLoRA and AdaLoRA), even without additional hyperparameter tuning. These results suggest that spectral clipping may facilitate more effective adaptation in spectral regions that are not fully exploited in prior approaches.

### P.5 Gram-Schmidt orthogonalization on parameterized singular vectors

Orthogonal regularization is a widely used method for implementing parameterized SVD in LoRA (Zhang et al., 2023; Cao, 2024; Zhang et al., 2024). Unlike orthogonal regularization, Gram-Schmidt orthogonalization—one of the orthogonalization methods—produces strictly orthogonal vectors. However, since it refines each subsequent vector based on the previously orthogonalized ones, it has the disadvantage of being difficult to parallelize compared to the regularization approach. Furthermore, as shown in Fig. 10, the orthogonal error rapidly decreases to below $10^{-2}$ in the early stages of fine-tuning, indicating that the singular vectors can be sufficiently trained. We conduct an ablation study with orthogonal regularization by replacing it with Gram-Schmidt-based orthogonalization. We measured the performance and training speed in Table 19.

### P.6 Ablation on layer-wise quantile

While the upper-bound quantile $q$ of the singular values can be applied globally across all layers, it is also possible to assign different quantiles to different groups of layers. To investigate this flexibility, we perform an ablation study in which we group the $L$ layers and assign distinct quantiles to each group. Specifically, in the 'Descending' setting, the $L$ layers are divided into four equal groups, and the quantile is assigned in decreasing order—from $q = 1.0$ to $q = 0.25$, which is quartile of pre-trained spectrum for simplicity—as the layer index increases. Conversely, in the 'Ascending' setting, the quantile assignment is reversed, increasing from $q = 0.25$ to $q = 1.0$. **SCLoRA** applies a single uniform quantile across all layers, which corresponds to our default design. As shown in Table 20 the relative performance differences across layer-wise quantile schedules are minor: the choice between Ascending and Descending varies slightly across datasets, and neither configuration leads to severe degradation. Importantly, the Global quantile consistently provides competitive or superior results, achieving the best performance on all four tasks. Considering both stability and simplicity, applying a single quantile remains the most effective and robust strategy.

| Method | CoLA | QNLI | MRPC | STS-B |
|---|---|---|---|---|
| LoRA | $64.49_{\pm 0.64}$ | $92.73_{\pm 0.11}$ | $89.05_{\pm 0.12}$ | $90.87_{\pm 0.08}$ |
| AdaLoRA | $62.41_{\pm 0.66}$ | $92.67_{\pm 0.12}$ | $89.38_{\pm 0.57}$ | $91.08_{\pm 0.11}$ |
| AdaLoRA+**SCLoRA** | $64.56_{\pm 0.12}$ | $92.79_{\pm 0.17}$ | $89.54_{\pm 0.99}$ | $91.08_{\pm 0.30}$ |
| MiLoRA | $64.31_{\pm 0.97}$ | $92.96_{\pm 0.21}$ | $89.30_{\pm 0.12}$ | $90.96_{\pm 0.05}$ |
| MiLoRA+**SCLoRA** | $64.57_{\pm 0.87}$ | $93.01_{\pm 0.11}$ | $89.30_{\pm 0.28}$ | $91.00_{\pm 0.04}$ |
| **SCLoRA** | $64.79_{\pm 0.20}$ | $93.09_{\pm 0.14}$ | $90.32_{\pm 0.86}$ | $91.22_{\pm 0.05}$ |

Table 18: Comparison of various methods on GLUE tasks with different random seeds.

| | CoLA | | RTE | | MRPC | | STS-B | |
|---|---|---|---|---|---|---|---|---|
| | Acc. | Min/Epoch | Acc. | Min/Epoch | Acc. | Min/Epoch | Acc. | Min/Epoch |
| Gram-Schmidt ortho. | $57.31_{\pm 0.85}$ | 3.6 | $67.27_{\pm 1.23}$ | 1.1 | $87.58_{\pm 1.02}$ | 2.0 | $89.88_{\pm 0.19}$ | 2.6 |
| **SCLoRA** | $\mathbf{64.79}_{\pm 0.20}$ | **2.9** | $\mathbf{83.15}_{\pm 0.17}$ | **0.8** | $\mathbf{90.32}_{\pm 0.86}$ | **1.3** | $\mathbf{91.22}_{\pm 0.05}$ | **2.1** |

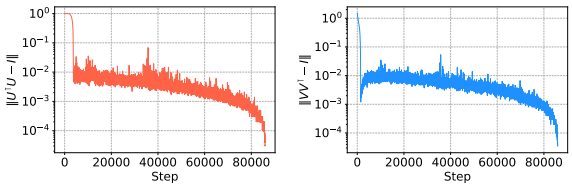
Table 19: Comparison of Gram-Schmidt orthogonalization and orthogonal regularization across various datasets. 'Min/Epoch' denotes the number of minutes required per epoch.

| Method | CoLA | RTE | MRPC | STS-B |
|---|---|---|---|---|
| Descending | 63.03 | 83.03 | 89.87 | 90.84 |
| Ascending | 63.40 | 80.99 | 88.64 | 91.06 |
| **SCLoRA** | **64.79** | **83.15** | **90.32** | **91.22** |

Table 20: Ablation study on layerwise $q$

| Method | CoLA | RTE | MRPC | STS-B |
|---|---|---|---|---|
| Learnable $\bar{\sigma}$ | 64.21 | 82.43 | 90.28 | 91.10 |
| **SCLoRA** | **64.79** | **83.15** | **90.32** | **91.22** |

Table 21: Ablation study on automatic $\bar{\sigma}$




(a) Orthogonal loss of $U$ (b) Orthogonal loss of $V^\top$

Figure 10: The orthogonal loss curves of parameterized singular vectors $U$ and $V$ when fine-tuning RoBERTa$_{\text{base}}$ on STS-B dataset

### P.7 Ablation on automatic $\bar{\sigma}$

Since **SCLoRA** uses a fixed quantile $q$, the corresponding spectral bound $\bar{\sigma}$ is also fixed. Instead of using this fixed value, one may alternatively treat $\bar{\sigma}$ as a learnable parameter so that it can be automatically learned during training. We conduct an ablation study where $\bar{\sigma}$ is made learnable, and the results are reported in Table 21. The results show that using a learnable $\bar{\sigma}$ does not lead to a substantial degradation in performance. However, the fixed $\bar{\sigma}$ generally yields more stable and consistently better results across tasks. In addition, since a learnable $\bar{\sigma}$ introduces extra parameters and potential computational overhead, we choose to use a fixed constant value rather than adopting the automatically learned variant.

## Q Orthogonal Regularization on Parameterized Singular Vectors

As shown in Fig. 3, when the $q$ and corresponding $\bar{\sigma}$ exceeds a certain level, the fine-tuning performance remains stable, and the performance on the pre-training task is largely recovered. In addition, we further treat the $\bar{\sigma}$ as a learnable variable and perform auto-tuning. As presented in Table 21, making the quantile learnable maintains stable overall performance, and no performance degradation is observed. However, under our experimental setup, the learnable $\bar{\sigma}$ setting does not yield clear additional gains over the fixed setting. For simplicity and consistency, we therefore adopt the fixed $\bar{\sigma}$ configuration as the default choice.

Fig. 10 shows the orthogonal loss curve of parameter singular vectors $U$ and $V$ of RoBERTa$_{\text{base}}$ fine-tuned on STS-B dataset. The singular vectors are orthogonally optimized as indicated by the consistent reduction in orthogonal loss.

## R Empirical Complexity Analysis

We conduct additional experiments on empirical time complexity and GPU usage during fine-tuning and inference phase. Table 22 summarizes the empirical training time (min per epoch) and peak GPU

| Method | MNLI | SST-2 | CoLA | QQP | QNLI | RTE | MRPC | STS-B |
|---|---|---|---|---|---|---|---|---|
| LoRA | 105.9/24.9 | 18.2/24.9 | 2.3/24.9 | 98.1/24.9 | 28.3/24.9 | 0.7/24.9 | 1.0/12.5 | 1.6/24.9 |
| PiSSA | 106.2/24.9 | 18.1/24.9 | 2.3/24.9 | 98.1/24.9 | 28.2/24.9 | 0.7/24.9 | 1.0/12.5 | 1.5/24.9 |
| AdaLoRA | 123.4/25.6 | 21.1/25.6 | 2.7/25.6 | 114.4/25.6 | 33.1/25.6 | 0.8/25.6 | 1.3/13.1 | 1.8/25.6 |
| MiLoRA | 106.0/24.9 | 18.1/24.9 | 2.3/24.9 | 98.1/24.9 | 28.2/24.9 | 0.7/24.9 | 1.0/12.5 | 1.5/24.9 |
| **SCLoRA** | 128.9/25.2 | 22.1/25.2 | 2.8/25.2 | 119.4/25.2 | 34.3/25.2 | 0.8/25.2 | 1.3/12.8 | 1.9/25.2 |

Table 22: Comparison of training time (min per epoch) and peak GPU usage (GB) with RoBERTa$_{\text{base}}$

| Method | MNLI | SST-2 | CoLA | QQP | QNLI | RTE | MRPC | STS-B |
|---|---|---|---|---|---|---|---|---|
| LoRA | 0.85/0.3 | 0.08/0.3 | 0.10/0.3 | 3.46/0.3 | 0.47/0.3 | 0.03/0.3 | 0.05/0.3 | 0.14/0.3 |
| PiSSA | 0.84/0.3 | 0.08/0.3 | 0.09/0.3 | 3.47/0.3 | 0.47/0.3 | 0.03/0.3 | 0.04/0.3 | 0.13/0.3 |
| AdaLoRA | 1.32/0.3 | 0.13/0.3 | 0.15/0.3 | 5.43/0.3 | 0.74/0.3 | 0.05/0.3 | 0.07/0.3 | 0.21/0.3 |
| MiLoRA | 0.84/0.3 | 0.08/0.3 | 0.09/0.3 | 3.47/0.3 | 0.47/0.3 | 0.03/0.3 | 0.04/0.3 | 0.13/0.3 |
| **SCLoRA** | 1.01/0.3 | 0.10/0.3 | 0.12/0.3 | 4.15/0.3 | 0.57/0.3 | 0.04/0.3 | 0.05/0.3 | 0.16/0.3 |

Table 23: Comparison of inference time (min per epoch) and peak GPU usage (GB) with RoBERTa$_{\text{base}}$

| Method | LLaMA-7B | LLaMA2-7B |
|---|---|---|
| LoRA | 74.7/0.53 | 77.6/0.54 |
| **SCLoRA** | 79.4/0.51 | 81.0/0.52 |

Table 24: Comparison of training time (min per epoch) and peak GPU usage (GB) with LLaMA family

usage (GB) of RoBERTa$_{\text{base}}$ fine-tuned on GLUE tasks. The GPU usage showed a very slight increase compared to the original LoRA, and the additional runtime occurs in **SCLoRA** and AdaLoRA. This increase arises from the orthogonal regularization of singular vectors generated by parameterized SVD. However, fine-tuning typically requires fewer epochs, and considering the improved performance and the ability to retain pre-trained knowledge compared to the baseline model, this increase is negligible. Table 23 reports the empirical inference time (min per epoch) and peak GPU usage (GB) of RoBERTa$_{\text{base}}$ fine-tuned on GLUE tasks. **SCLoRA** exhibits a marginal increase in inference latency and peak GPU memory relative to vanilla LoRA; nonetheless, both metrics remain lower than those of AdaLoRA and are practically comparable. In large models such as LLaMA, the primary computational complexity stems from the high dimensionality of the base model. For instance, we empirically measure the computational complexity in LLaMA-7B and LLaMA2-7B, and we report "average accuracy on commonsense reasoning / the steps per second during fine-tuning" in Table 24. Since $r \ll d$, the training process runs at approximately 96% of the original speed, which is not a significant difference.