SDS-LoRA: Overcoming Anisotropic Gradient Scaling in Low-Rank Adaptation

Junghun Oh[*]
Dept. of ECE, ASRI
Seoul National University
Nota Inc.
dh6dh@snu.ac.kr

Sungyong Baik[†]
Dept. of Artificial Intelligence
Dept. of Data Science
Hanyang University
dsybaik@hanyang.ac.kr

Kyoung Mu Lee
Dept. of ECE, ASRI, IPAI
Seoul National University
kyoungmu@snu.ac.kr

Abstract

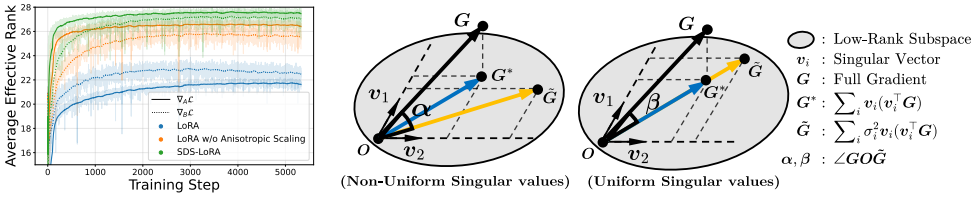
Low-Rank Adaptation (LoRA) enables efficient adaptation of large pretrained models to downstream tasks by parameterizing weight updates with low-rank matrices. In this paper, we investigate the limitations of the LoRA parameterization from a geometric perspective. Specifically, we show that when a full fine-tuning gradient is backpropagated to the low-rank matrices, it undergoes anisotropic scaling driven by their singular values. We argue that this phenomenon is undesirable because it distorts the full fine-tuning gradient by skewing it toward dominant singular directions while suppressing others. Our analyses demonstrate that anisotropic gradient scaling reduces the effective rank of the low-rank matrices' gradients and fails to provide the best possible alignment between the full fine-tuning gradient and its low-rank approximation in LoRA for arbitrary gradients, thereby exacerbating the gap to full fine-tuning. To address these limitations, we propose a new low-rank parameterization, SDS-LoRA, which **S**tructurally **D**ecouples **S**ingular values from the backward pass. Our method ensures that the full fine-tuning gradient backpropagates only through the orthonormal bases of the low-rank matrices' subspaces, independent of their scales. Convergence analysis demonstrates that while LoRA's convergence rate degrades with the condition number of the low-rank matrices, that of SDS-LoRA remains independent of it. Experimental results across natural language and vision benchmarks show that SDS-LoRA improves loss convergence and reduces the gap to full fine-tuning, significantly enhancing adaptation performance. The code is available at https://github.com/JungHunOh/SDS-LoRA_NeurIPS2026.

1 Introduction

Large-scale pretrained Transformer models [47], such as GPT [40], LLaMA [46], and ViTs [14], have achieved remarkable success by learning rich representations from massive datasets. In recent years, adapting these versatile models to a wide range of downstream tasks has gained immense popularity. A few studies have explored improving task adaptation through adaptive optimization strategies [3, 4] and also investigated fine-tuning strategies to mitigate parameter interference across task-specific models [28]. While this pretraining-and-adaptation paradigm has become the standard in the field, full fine-tuning of these models is often computationally expensive and impractical. For example, GPT-3 [7] has approximately 175 billion parameters, and LLaMA-3 variants [15] range from 8 to 405 billion parameters. To address this, Low-Rank Adaptation (LoRA) [19] leverages the observation that model updates reside in a low-rank subspace [1, 29], representing weight updates through trainable low-rank matrices: $\Delta \boldsymbol{W} = \boldsymbol{BA}$. Consequently, LoRA drastically reduces the number of trainable parameters.

[*]This work was done at Seoul National University.
[†]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Effective rank of gradients (b) Alignment between the full gradient $\boldsymbol{G}$ and its approximation $\tilde{\boldsymbol{G}}$

Figure 1: **Adverse effects of anisotropic gradient scaling in LoRA. (a)** At each step of LoRA training, we compare the effective rank of the original gradients of the LoRA matrices (blue) to the modified gradients where anisotropic gradient scaling induced by singular values has been removed (orange). We observe a substantial reduction in effective rank when gradients are scaled by singular values. In contrast, SDS-LoRA overcomes anisotropic gradient scaling, thereby maintaining a higher effective gradient rank (green). **(b)** Non-uniform singular values of LoRA matrices cause the effective gradient with respect to $\Delta\boldsymbol{W}$ ($\tilde{\boldsymbol{G}}$) to misalign with the orthogonal projection of $\boldsymbol{G}$ onto the low-rank subspace ($\boldsymbol{G}^*$) by applying anisotropic scaling to $\boldsymbol{G}^*$. This results in suboptimal alignment between $\boldsymbol{G}$ and $\tilde{\boldsymbol{G}}$ ($\alpha > \beta$).

Although LoRA has attracted significant attention for its efficiency, a substantial performance gap remains relative to full fine-tuning, motivating further improvements to the LoRA framework. In this paper, we investigate the limitations of the LoRA parameterization from a geometric perspective. Specifically, we note that the optimization dynamics in LoRA are adversely affected by the singular-value structure of the LoRA matrices (i.e., $\boldsymbol{A}$ and $\boldsymbol{B}$). For example, when calculating the gradient with respect to $\boldsymbol{A}$, the gradient with respect to $\Delta\boldsymbol{W}$, which we refer to as the full (fine-tuning) gradient, is backpropagated through $\boldsymbol{B}$. In this process, the singular values of $\boldsymbol{B}$ strengthen directions associated with large singular values while relatively diminishing others. This phenomenon may be undesirable because it distorts the full gradient by biasing it toward dominant singular directions, thereby concentrating its rich information on only a few directions. Figure 1a demonstrates that the effective rank of the LoRA matrices' gradients is significantly reduced due to the anisotropic scaling, potentially limiting LoRA's learning capacity. Furthermore, we theoretically show that uniform singular values uniquely maximize the worst-case quality with which the effective gradient with respect to $\Delta\boldsymbol{W}$ approximates the full gradient, whereas anisotropic gradient scaling degrades it (Figure 1b), causing the optimization dynamics of LoRA to deviate from those of full fine-tuning.

To overcome the adverse effects of anisotropic gradient scaling, we propose a new parameterization for $\Delta\boldsymbol{W}$, named SDS-LoRA, that **S**tructurally **D**ecouples the **S**ingular values of the LoRA matrices from the backward pass. As illustrated in Figure 2, SDS-LoRA multiplies each LoRA matrix by a matrix representing the orthonormal bases of the other's subspace (i.e., $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$) and sums the two products. By treating $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$ as constants in the backward pass, this formulation ensures that the singular values of the LoRA matrices do not participate in the gradient computation, thereby eliminating anisotropic gradient scaling. Consequently, as shown in Figure 1a, our method results in a significantly higher effective rank for the gradients of $\boldsymbol{A}$ and $\boldsymbol{B}$ compared to LoRA. Moreover, as shown by the uniform-singular-value case in Figure 1b, our method yields an effective gradient with respect to $\Delta\boldsymbol{W}$ that nearly attains this worst-case optimum, thereby reducing the gap to full fine-tuning. Our theoretical analysis proves that, while the convergence rate of LoRA is degraded by the condition number of the LoRA matrices, that of our method remains independent of it. Experiments across various benchmarks and models demonstrate that SDS-LoRA significantly improves loss convergence and narrows the gap to full fine-tuning, outperforming existing methods.

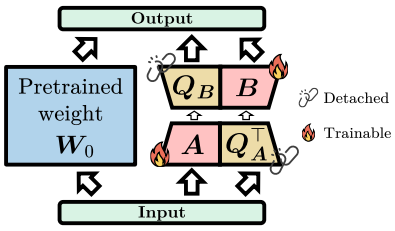


Figure 2: **Illustration of SDS-LoRA**

We summarize the contribution of our work as follows:

- We provide a novel perspective on the limitations of LoRA, showing that LoRA suffers from anisotropic gradient scaling induced by the singular values of the LoRA matrices.
- We demonstrate that anisotropic gradient scaling reduces the effective rank of the LoRA matrices' gradients and does not attain the best approximation of arbitrary full fine-tuning gradients.
- We propose a new parameterization, SDS-LoRA, that overcomes anisotropic gradient scaling in LoRA by structurally decoupling the singular values from the backward pass.

- SDS-LoRA substantially improves loss convergence and narrows the gap to full fine-tuning, outperforming LoRA and existing methods.

2 Related Work

LoRA Variants. Given the success of LoRA, numerous follow-up studies have been conducted to further enhance its performance [23, 17, 43]. Several works propose alternative formulations of low-rank weight updates to further improve the performance–efficiency trade-off [35, 26, 33, 2, 25]. Other research directions include developing effective initialization strategies for the trainable low-rank matrices [16, 49, 37, 61, 30], improving the rank of weight updates [21, 32], and designing layer-wise adaptive rank search methods [59, 24]. Another line of work leverages orthogonalization: OLoRA [8] initializes the LoRA matrices from a QR decomposition of the pretrained weight, and QR-LoRA [53] freezes an orthonormal factor and trains only the triangular one. The former restores isotropy only at initialization, while the latter confines the row space of $\Delta \boldsymbol{W}$ to a subspace determined by the pretrained weight, sacrificing representational capacity. See Section I for a mechanism-level and empirical comparison.

Full Fine-Tuning Approximation Perspective. Several recent studies have examined LoRA from the perspective of approximating the full fine-tuning gradient [49, 61, 50, 58, 56, 45]. Wang et al. [49] and Zhang et al. [61] propose initialization strategies for LoRA that minimize the gradient approximation error at the first training step. Several recent works attempt to minimize the gap between the full fine-tuning gradient and the effective gradient with respect to weight updates via preconditioning [50, 58, 56, 45, 54, 60]. Specifically, they scale the gradient of each LoRA matrix by the inverse Gram matrix of the other LoRA matrix. While these approaches eliminate the influence of singular values on the effective gradient, anisotropic gradient scaling persists within the LoRA matrices' gradients. Moreover, because these methods modify the original gradients, they require additional mechanisms to adjust momentum, which hinders their practical adoption. See Section H for a detailed comparison to the preconditioning-based methods.

Learning in Stiefel Manifolds. Li et al. [31] and Lion et al. [34] formulate weight updates as $\Delta \boldsymbol{W} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are constrained to Stiefel manifolds and $\boldsymbol{S}$ is a scale matrix. These methods also modify the Euclidean gradient to produce a Riemannian gradient and require additional retraction or regularization to ensure that $\boldsymbol{U}$ and $\boldsymbol{V}$ remain on the manifolds. Importantly, the updates for $\boldsymbol{U}$ and $\boldsymbol{V}$ depend on the scale matrix $\boldsymbol{S}$, failing to overcome anisotropic gradient scaling. Related studies have also employed SVD-based low-rank subspace representations for task identification and routing [22, 44].

Unlike prior works, we focus on the adverse effects of anisotropic gradient scaling, offering a novel perspective on LoRA's limitations. Our method overcomes these effects by structurally eliminating the scaling effects from the backward pass without modifying the original gradients.

3 Proposed Method

3.1 Preliminaries

Let $\boldsymbol{W}_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ denote the weight matrix of a linear layer in a pretrained model, with input dimension d_{in} and output dimension d_{out}. We consider fine-tuning $\boldsymbol{W}_0$ to adapt the model to a downstream task. Instead of updating all the parameters of $\boldsymbol{W}_0$, Low-Rank Adaptation (LoRA) assumes a low-rank structure for the desired weight update [1, 29] and learns a low-rank decomposition of the update:

$$\boldsymbol{W}_{\text{eff}} = \boldsymbol{W}_0 + \Delta \boldsymbol{W}_{\text{LoRA}} = \boldsymbol{W}_0 + s\boldsymbol{B}\boldsymbol{A}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{r \times d_{\text{in}}}$ and $\boldsymbol{B} \in \mathbb{R}^{d_{\text{out}} \times r}$ are trainable LoRA matrices of rank r, and s is a constant scaling factor. For a single layer, the total number of trainable parameters in LoRA is $r \times (d_{\text{out}} + d_{\text{in}})$. Assuming $r \ll \min(d_{\text{out}}, d_{\text{in}})$, this is significantly smaller than the $d_{\text{out}} \times d_{\text{in}}$ parameters required in a full fine-tuning scenario.

Let $\boldsymbol{G} = \nabla_{\boldsymbol{W}_{\text{eff}}} \mathcal{L}$ denote the gradient of the loss $\mathcal{L}$ with respect to the effective weights. We refer to $\boldsymbol{G}$ as the *full gradient*, as it is used in full fine-tuning. By the chain rule, the gradients with respect to the LoRA matrices are given by $\nabla_{\boldsymbol{A}} \mathcal{L} = s\boldsymbol{B}^\top \boldsymbol{G}$, $\nabla_{\boldsymbol{B}} \mathcal{L} = s\boldsymbol{G}\boldsymbol{A}^\top$. Let $\boldsymbol{A} = \boldsymbol{U}_{\boldsymbol{A}} \boldsymbol{\Sigma}_{\boldsymbol{A}} \boldsymbol{V}_{\boldsymbol{A}}^\top$ and $\boldsymbol{B} = \boldsymbol{U}_{\boldsymbol{B}} \boldsymbol{\Sigma}_{\boldsymbol{B}} \boldsymbol{V}_{\boldsymbol{B}}^\top$ be the rank-r truncated singular value decompositions (SVD) of the LoRA

matrices, where $\boldsymbol{V_A} \in \mathbb{R}^{d_{\text{in}} \times r}$ and $\boldsymbol{U_B} \in \mathbb{R}^{d_{\text{out}} \times r}$ have orthonormal columns, $\boldsymbol{\Sigma_A}, \boldsymbol{\Sigma_B} \in \mathbb{R}^{r \times r}$ are the diagonal matrices of singular values, and $\boldsymbol{U_A}, \boldsymbol{V_B} \in \mathbb{R}^{r \times r}$ are orthogonal matrices. Substituting the SVD representations into the gradient equations, we obtain

$$\nabla_{\boldsymbol{A}} \mathcal{L} = s \boldsymbol{V_B} \boldsymbol{\Sigma_B} \boldsymbol{U_B}^\top \boldsymbol{G}, \;\; \nabla_{\boldsymbol{B}} \mathcal{L} = s \boldsymbol{G} \boldsymbol{V_A} \boldsymbol{\Sigma_A} \boldsymbol{U_A}^\top \quad \text{(in LoRA)}. \tag{2}$$

3.2 Adverse Effects of Anisotropic Gradient Scaling via Singular Values

We provide a geometric interpretation of how the full gradient $\boldsymbol{G}$ is backpropagated to yield $\nabla_{\boldsymbol{A}} \mathcal{L}$ and $\nabla_{\boldsymbol{B}} \mathcal{L}$. Equation 2 shows that $\boldsymbol{G}$ undergoes the following transformations:

- $\boldsymbol{V_A}$ and $\boldsymbol{U_B}^\top$: Acting as partial isometries, these matrices perform isometric projections of $\boldsymbol{G}$ into their respective subspaces.
- $\boldsymbol{\Sigma_A}$ and $\boldsymbol{\Sigma_B}$: If the singular values of $\boldsymbol{A}$ and $\boldsymbol{B}$ are non-uniform, these matrices induce anisotropic scaling that skews the projected gradient toward the dominant singular directions.
- $\boldsymbol{U_A}^\top$ and $\boldsymbol{V_B}$: These matrices perform isometric transformations.

Note that the isometric projection by $\boldsymbol{V_A}$ and $\boldsymbol{U_B}^\top$ is an inherent bottleneck of low-rank adaptation, and $\boldsymbol{U_A}^\top$ and $\boldsymbol{V_B}$ act merely as isometries, preserving the geometric properties of the projected gradient. **However, the anisotropic scaling through $\boldsymbol{\Sigma_A}$ and $\boldsymbol{\Sigma_B}$ may be undesirable because it makes the backpropagated gradient depend on which directions the spectrum happens to emphasize, leaving the remaining directions with an arbitrarily small share of the full gradient in the worst case.**

As discussed in Section F, we observe that the stable ranks of $\boldsymbol{A}$ and $\boldsymbol{B}$ are substantially lower than the LoRA rank r, suggesting that their singular values are highly skewed. To demonstrate the adverse effects of anisotropic gradient scaling resulting from these skewed singular values, we compare the effective rank of the original gradients (i.e., $s\boldsymbol{V_B}\boldsymbol{\Sigma_B}\boldsymbol{U_B}^\top\boldsymbol{G}$ and $s\boldsymbol{G}\boldsymbol{V_A}\boldsymbol{\Sigma_A}\boldsymbol{U_A}^\top$) with that of modified gradients that are independent of singular values (i.e., $s\boldsymbol{V_B}\boldsymbol{U_B}^\top\boldsymbol{G}$ and $s\boldsymbol{G}\boldsymbol{V_A}\boldsymbol{U_A}^\top$). Figure 1a shows that the effective rank is significantly reduced when the gradients are scaled by the singular values. These results suggest that anisotropic gradient scaling concentrates the rich information of $\boldsymbol{G}$ on a few dominant directions, potentially limiting LoRA's learning capacity.

Furthermore, anisotropic gradient scaling causes the optimization dynamics of LoRA to markedly deviate from those of full fine-tuning. To examine this, we first derive the effective gradient with respect to $\boldsymbol{W}_{\text{eff}}$. Under gradient descent with learning rate η, the change in $\boldsymbol{W}_{\text{eff}}$ resulting from updates to $\boldsymbol{A}$ and $\boldsymbol{B}$ is given by $\Delta \boldsymbol{W}_{\text{eff}} \approx s(\Delta \boldsymbol{B}\boldsymbol{A} + \boldsymbol{B}\Delta\boldsymbol{A}) = -\eta s^2(\boldsymbol{G}\boldsymbol{A}^\top\boldsymbol{A} + \boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{G})$. Substituting the SVD of $\boldsymbol{A}$ and $\boldsymbol{B}$ into this, the effective gradient with respect to $\boldsymbol{W}_{\text{eff}}$ is expressed as

$$\tilde{\boldsymbol{G}} = s^2(\boldsymbol{G}\boldsymbol{V_A}\boldsymbol{\Sigma_A}^2\boldsymbol{V_A}^\top + \boldsymbol{U_B}\boldsymbol{\Sigma_B}^2\boldsymbol{U_B}^\top\boldsymbol{G}) \quad \text{(in LoRA)}. \tag{3}$$

Consequently, LoRA can be interpreted as full fine-tuning with $\tilde{\boldsymbol{G}}$, a low-rank approximation of $\boldsymbol{G}$ [49, 50, 58, 56, 45]. We analyze how the gap between $\boldsymbol{G}$ and $\tilde{\boldsymbol{G}}$ influences the optimization dynamics through the descent lemma [39]. Assuming that the loss function $\mathcal{L}$ is β-smooth with respect to $\boldsymbol{W}_{\text{eff}}$, we obtain the following loss decrease bound when $\boldsymbol{W}_{\text{eff}}$ is updated by $\tilde{\boldsymbol{G}}$:

$$\mathcal{L}(\boldsymbol{W}_{\text{eff}} - \eta\tilde{\boldsymbol{G}}) - \mathcal{L}(\boldsymbol{W}_{\text{eff}}) \leq -\eta\langle \boldsymbol{G}, \tilde{\boldsymbol{G}}\rangle_F + \frac{\beta}{2}\eta^2||\tilde{\boldsymbol{G}}||_F^2, \tag{4}$$

where $\langle\cdot,\cdot\rangle_F$ and $||\cdot||_F$ denote the Frobenius inner product and norm, respectively. Equation 4 suggests that, for a sufficiently small η, the loss decrease bound is degraded as $\tilde{\boldsymbol{G}}$ diverges from $\boldsymbol{G}$ in terms of the Frobenius inner product. As such, we consider $\langle \boldsymbol{G}, \tilde{\boldsymbol{G}}\rangle_F$ as a measure of the approximation quality of $\tilde{\boldsymbol{G}}$. Note that $\boldsymbol{A}$ and $\boldsymbol{B}$ persist throughout training and encounter a different full gradient $\boldsymbol{G}$ at every step. A desirable singular value configuration must therefore remain effective for arbitrary $\boldsymbol{G}$ rather than being tuned to a single instance, which motivates a worst-case criterion over $\boldsymbol{G}$. To formalize this, define $\boldsymbol{M_A} = \boldsymbol{V_A}^\top\boldsymbol{G}^\top\boldsymbol{G}\boldsymbol{V_A}$ and $\boldsymbol{M_B} = \boldsymbol{U_B}^\top\boldsymbol{G}\boldsymbol{G}^\top\boldsymbol{U_B}$, so that the gradient approximation quality in LoRA is written as

$$\langle \boldsymbol{G}, \tilde{\boldsymbol{G}}\rangle_F = s^2 J, \quad \text{where} \quad J = \mathrm{Tr}(\boldsymbol{M_A}\boldsymbol{\Sigma_A}^2) + \mathrm{Tr}(\boldsymbol{M_B}\boldsymbol{\Sigma_B}^2).$$

A worst-case value of J alone, however, is not comparable across different $\boldsymbol{G}$, since J is quadratic in $\boldsymbol{G}$ and an adversarial choice could simply take a $\boldsymbol{G}$ of small norm. We therefore compare J against

the best value achievable within the given subspaces for that same $\boldsymbol{G}$, $J^\star(\boldsymbol{G}) = E_{\boldsymbol{A}}\lambda_{\max}(\boldsymbol{M}_{\boldsymbol{A}}) + E_{\boldsymbol{B}}\lambda_{\max}(\boldsymbol{M}_{\boldsymbol{B}})$ with $E_{\boldsymbol{A}} = ||\boldsymbol{A}||_F^2$ and $E_{\boldsymbol{B}} = ||\boldsymbol{B}||_F^2$, and evaluate a configuration by the scale-invariant worst-case ratio $\min_{\boldsymbol{G}} J/J^\star(\boldsymbol{G})$, where the minimum is over $\boldsymbol{G}$ with $J^\star(\boldsymbol{G}) > 0$.

Theorem 3.1. *Assume that the row space of $\boldsymbol{A}$ and the column space of $\boldsymbol{B}$ are given, together with the energies $E_{\boldsymbol{A}}$ and $E_{\boldsymbol{B}}$, so that the admissible configurations are those satisfying $\mathrm{Tr}(\boldsymbol{\Sigma}_{\boldsymbol{A}}^2) = E_{\boldsymbol{A}}$ and $\mathrm{Tr}(\boldsymbol{\Sigma}_{\boldsymbol{B}}^2) = E_{\boldsymbol{B}}$. Then, $\boldsymbol{\Sigma}_{\boldsymbol{A}} = \sqrt{E_{\boldsymbol{A}}/r}\boldsymbol{I}_r$ and $\boldsymbol{\Sigma}_{\boldsymbol{B}} = \sqrt{E_{\boldsymbol{B}}/r}\boldsymbol{I}_r$ uniquely maximize the worst-case ratio $\min_{\boldsymbol{G}} J/J^\star(\boldsymbol{G})$, attaining the optimal value $\frac{1}{r}$.*

See Section A for the proof. Theorem 3.1 suggests that, under the same subspace and energy configurations, uniform singular values are the unique configuration that guarantees the best gradient approximation quality for arbitrary $\boldsymbol{G}$. A non-uniform configuration, in contrast, is favorable only for the gradient directions that its spectrum happens to emphasize: for a $\boldsymbol{G}$ concentrated on the smallest singular directions, its ratio degrades to $\frac{\lambda_{\min}(\boldsymbol{\Sigma}_{\boldsymbol{A}}^2)+\lambda_{\min}(\boldsymbol{\Sigma}_{\boldsymbol{B}}^2)}{E_{\boldsymbol{A}}+E_{\boldsymbol{B}}}$ (Equation 11), which falls strictly below $\frac{1}{r}$ and vanishes as the spectrum becomes more skewed. Under the optimal condition, $\tilde{\boldsymbol{G}}$ is simplified to $s^2(\frac{E_{\boldsymbol{A}}}{r}\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} + \frac{E_{\boldsymbol{B}}}{r}\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})}\boldsymbol{G})$ where $\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}$ and $\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})}$ denote the orthogonal projection matrices onto the row space of $\boldsymbol{A}$ ($\mathcal{R}(\boldsymbol{A})$) and the column space of $\boldsymbol{B}$ ($\mathcal{C}(\boldsymbol{B})$), respectively. As illustrated in Figure 1b, while the non-uniform singular values distort the projected gradients, the uniform singular values scale the projection equally in all directions, resulting in the best worst-case alignment between $\tilde{\boldsymbol{G}}$ and $\boldsymbol{G}$.

3.3 SDS-LoRA: Structural Decoupling of Singular Values from Backward Pass

Our analysis in Section 3.2 demonstrates the negative effects of non-uniform singular values of $\boldsymbol{A}$ and $\boldsymbol{B}$ during the backward pass. However, addressing this limitation is not straightforward, as the roles of singular values in the forward and backward passes conflict. In the forward pass, the singular values play a crucial role in representing $\Delta\boldsymbol{W}_{\text{LoRA}}$. Enforcing uniform singular values of $\boldsymbol{A}$ and $\boldsymbol{B}$ significantly restricts the class of representable weight updates.[3] Conversely, allowing fully expressive weight updates conflicts with the uniform singular value condition for $\boldsymbol{A}$ and $\boldsymbol{B}$.

To address this, we propose a new low-rank parameterization of weight updates, named SDS-LoRA, which **S**tructurally **D**ecouples the **S**ingular values of LoRA matrices from the backward pass while preserving their role in the forward pass for representing weight updates. To do so, we utilize the orthonormal bases of $\mathcal{R}(\boldsymbol{A})$ and $\mathcal{C}(\boldsymbol{B})$. Specifically, we perform QR decomposition of $\boldsymbol{A}^\top$ and $\boldsymbol{B}$: $\boldsymbol{A}^\top = \boldsymbol{Q}_{\boldsymbol{A}}\boldsymbol{R}_{\boldsymbol{A}}$ and $\boldsymbol{B} = \boldsymbol{Q}_{\boldsymbol{B}}\boldsymbol{R}_{\boldsymbol{B}}$ where the columns of $\boldsymbol{Q}_{\boldsymbol{A}}$ and $\boldsymbol{Q}_{\boldsymbol{B}}$ form orthonormal bases of $\mathcal{R}(\boldsymbol{A})$ and $\mathcal{C}(\boldsymbol{B})$, respectively. We then formulate the weight updates as follows:

$$\Delta\boldsymbol{W}_{\text{SDS-LoRA}} = s(\boldsymbol{Q}_{\boldsymbol{B}}\boldsymbol{A} + \boldsymbol{B}\boldsymbol{Q}_{\boldsymbol{A}}^\top) = s[\boldsymbol{Q}_{\boldsymbol{B}}\ \boldsymbol{B}]\begin{bmatrix}\boldsymbol{A}\\ \boldsymbol{Q}_{\boldsymbol{A}}^\top\end{bmatrix}. \tag{5}$$

Importantly, we treat $\boldsymbol{Q}_{\boldsymbol{A}}$ and $\boldsymbol{Q}_{\boldsymbol{B}}$ as constants during the backward pass. This ensures that the LoRA matrices receive gradients only through $\boldsymbol{Q}_{\boldsymbol{A}}$ and $\boldsymbol{Q}_{\boldsymbol{B}}$, while also making training more efficient by eliminating the costly backpropagation through the QR decomposition. Consequently, the gradients with respect to $\boldsymbol{A}$ and $\boldsymbol{B}$ are expressed as

$$\nabla_{\boldsymbol{A}}\mathcal{L} = s\boldsymbol{Q}_{\boldsymbol{B}}^\top\boldsymbol{G},\ \nabla_{\boldsymbol{B}}\mathcal{L} = s\boldsymbol{G}\boldsymbol{Q}_{\boldsymbol{A}} \quad \text{(in SDS-LoRA)}. \tag{6}$$

Equation 6 shows that **the full gradient $\boldsymbol{G}$ is backpropagated to the LoRA matrices only through the partial isometries $\boldsymbol{Q}_{\boldsymbol{A}}$ and $\boldsymbol{Q}_{\boldsymbol{B}}$, eliminating the negative effects of anisotropic scaling via singular values.** This can be achieved without compromising the role of singular values in the forward pass, allowing $\Delta\boldsymbol{W}_{\text{SDS-LoRA}}$ to represent any rank-r matrix.

As shown in Figure 1a, SDS-LoRA overcomes the adverse effects of anisotropic gradient scaling, resulting in a higher effective rank in the gradients of $\boldsymbol{A}$ and $\boldsymbol{B}$ compared to LoRA. Moreover, SDS-LoRA narrows the gap between the full gradient $\boldsymbol{G}$ and its low-rank approximation $\tilde{\boldsymbol{G}}$. To examine this, we derive the change in the effective weight under the SDS-LoRA formulation, $\boldsymbol{W}_{\text{eff}} = \boldsymbol{W}_0 + \Delta\boldsymbol{W}_{\text{SDS-LoRA}}$. Although $\boldsymbol{Q}_{\boldsymbol{A}}$ and $\boldsymbol{Q}_{\boldsymbol{B}}$ are treated as constants during the backward

[3] If both $\boldsymbol{A}$ and $\boldsymbol{B}$ have uniform singular values, $\Delta\boldsymbol{W}_{\text{LoRA}}$ also has uniform singular values. If $\boldsymbol{A}$ has uniform singular values, then $\boldsymbol{A} = c\boldsymbol{U}_{\boldsymbol{A}}\boldsymbol{V}_{\boldsymbol{A}}^\top$ for some $c > 0$, and similarly $\boldsymbol{B} = d\boldsymbol{U}_{\boldsymbol{B}}\boldsymbol{V}_{\boldsymbol{B}}^\top$ for some $d > 0$. Then $\Delta\boldsymbol{W}_{\text{LoRA}} = \boldsymbol{B}\boldsymbol{A} = cd\boldsymbol{U}_{\boldsymbol{B}}(\boldsymbol{V}_{\boldsymbol{B}}^\top\boldsymbol{U}_{\boldsymbol{A}})\boldsymbol{V}_{\boldsymbol{A}}^\top$. Since $\boldsymbol{V}_{\boldsymbol{B}}^\top\boldsymbol{U}_{\boldsymbol{A}}$ is a product of orthogonal matrices and hence orthogonal, both $\boldsymbol{U}_{\boldsymbol{B}}(\boldsymbol{V}_{\boldsymbol{B}}^\top\boldsymbol{U}_{\boldsymbol{A}})$ and $\boldsymbol{V}_{\boldsymbol{A}}$ have orthonormal columns, so all singular values of $\Delta\boldsymbol{W}_{\text{LoRA}}$ equal cd.

pass, we periodically update them to reflect updates in the LoRA matrices, as described in the Training Procedure paragraph below. Taking into account the variations in $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$, the resulting change in $\boldsymbol{W}_{\text{eff}}$ is given by $\Delta \boldsymbol{W}_{\text{eff}} \approx s(\boldsymbol{Q_B}\Delta\boldsymbol{A} + \Delta\boldsymbol{B}\boldsymbol{Q_A}^\top + \Delta\boldsymbol{Q_B}\boldsymbol{A} + \boldsymbol{B}\Delta\boldsymbol{Q_A}^\top) = -\eta s^2(\boldsymbol{G}\boldsymbol{Q_A}\boldsymbol{Q_A}^\top + \boldsymbol{Q_B}\boldsymbol{Q_B}^\top\boldsymbol{G}) + s(\Delta\boldsymbol{Q_B}\boldsymbol{A} + \boldsymbol{B}\Delta\boldsymbol{Q_A}^\top)$. Consequently, the effective gradient with respect to $\boldsymbol{W}_{\text{eff}}$ is expressed as

$$\tilde{\boldsymbol{G}} = s^2(\boldsymbol{G}\boldsymbol{Q_A}\boldsymbol{Q_A}^\top + \boldsymbol{Q_B}\boldsymbol{Q_B}^\top\boldsymbol{G}) + \boldsymbol{E} \quad \text{(in SDS-LoRA)}, \tag{7}$$

where $\boldsymbol{E} = -\frac{s}{\eta}(\Delta\boldsymbol{Q_B}\boldsymbol{A} + \boldsymbol{B}\Delta\boldsymbol{Q_A}^\top)$ collects the terms arising from the change in $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$. Because $\boldsymbol{Q_A}\boldsymbol{Q_A}^\top = \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}$ and $\boldsymbol{Q_B}\boldsymbol{Q_B}^\top = \boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})}$, our method attains the worst-case-optimal form of the gradient approximation characterized in Theorem 3.1, up to the error described by $\boldsymbol{E}$. In Section E, we show that the contribution of $\boldsymbol{E}$ is small: the ratio $||\boldsymbol{E}||_F/||\tilde{\boldsymbol{G}}||_F$ averages 0.09, 0.05, and 0.03 in the early, middle, and late phases of training, respectively (Table 10). Consistently, $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$ change slowly, as the cosine similarity between $\boldsymbol{Q_A}$ and $\boldsymbol{Q_A} + \Delta\boldsymbol{Q_A}$ (or $\boldsymbol{Q_B}$ and $\boldsymbol{Q_B} + \Delta\boldsymbol{Q_B}$) remains close to 1 except for the first few iterations. Thus, our method closely approaches the optimal $\tilde{\boldsymbol{G}}$, substantially reducing the gap to full fine-tuning.

Training Procedure. Following the standard initialization of $\boldsymbol{A}$ and $\boldsymbol{B}$, we sample $\boldsymbol{A}$ from a uniform distribution and set $\boldsymbol{B} = 0$. Consequently, $\boldsymbol{Q_B}$ cannot be defined from $\boldsymbol{B}$ at the beginning of training. To address this, we perform a warm-up training of T iterations using the standard LoRA formulation. After the warm-up, we compute the rank-r truncated SVD of the weight updates, $\Delta\boldsymbol{W}_{\text{LoRA}} = \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top$, and reparameterize it using the SDS-LoRA formulation by setting the LoRA matrices and orthonormal bases as follows:

$$\boldsymbol{A} = \frac{1}{2s}\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top,\ \boldsymbol{B} = \frac{1}{2s}\boldsymbol{U}_r\boldsymbol{\Sigma}_r,\ \boldsymbol{Q_A} = \boldsymbol{V}_r,\ \boldsymbol{Q_B} = \boldsymbol{U}_r. \tag{8}$$

The coefficients in Equation 8 are chosen so that the reparameterization exactly reproduces the warm-up result: substituting them into Equation 5 gives $s(\boldsymbol{Q_B}\boldsymbol{A} + \boldsymbol{B}\boldsymbol{Q_A}^\top) = s(\frac{1}{2s}\boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top + \frac{1}{2s}\boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top) = \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top = \Delta\boldsymbol{W}_{\text{LoRA}}$. Using $\boldsymbol{\Sigma}_r^{1/2}$ instead would not reproduce this warm-up result, since Equation 5 sums $\boldsymbol{Q_B}\boldsymbol{A}$ and $\boldsymbol{B}\boldsymbol{Q_A}^\top$ rather than multiplying $\boldsymbol{A}$ and $\boldsymbol{B}$ directly. We reset all optimizer states after this process to ensure that the warm-up stage does not affect the subsequent main training stage. During the main training stage, we recompute $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$ as $\boldsymbol{A}$ and $\boldsymbol{B}$ evolve. Although QR decomposition introduces minor computational overhead, we mitigate this cost by updating $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$ intermittently. To account for the larger learning rates early in training, we update them more frequently in the initial phase. Specifically, at the t-th iteration, we update them if $t \bmod \left\lceil \frac{kt}{T_{\text{total}}} \right\rceil = 0$ where k is a hyperparameter and T_{total} indicates the total number of iterations. This means that training is divided into k phases, with the update interval in the i-th phase set to i. See Algorithm 1 for the overall training procedure.

3.4 Convergence Analysis

In Section 3.2, we utilize the descent lemma to formally characterize the adverse effects of anisotropic gradient scaling. To demonstrate how overcoming this issue improves loss convergence, we extend this analysis to compare the convergence rates of LoRA and SDS-LoRA.

We first assume that the loss function with respect to $\boldsymbol{W}_{\text{eff}}$ satisfies the Polyak–Łojasiewicz (PL) condition, a standard assumption in non-convex optimization used to establish linear convergence rates: $\|\nabla_{\boldsymbol{W}_{\text{eff}}}\mathcal{L}\|_F^2 \geq 2\mu(\mathcal{L}(\boldsymbol{W}_{\text{eff}}) - \mathcal{L}^*)$ where $\mu > 0$ and $\mathcal{L}^*$ denotes the minimum loss. While this condition can be restrictive in general deep learning contexts, it is plausible when fine-tuning from a pretrained model, where the loss landscape is typically better conditioned [52]. Accordingly, we assume that the PL condition holds in a neighborhood of the pretrained weight $\boldsymbol{W}_0$. We also assume that the change in $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$ is negligible, resulting in $\boldsymbol{E} \approx 0$ in Equation 7. See Section E for empirical validation of this assumption. Additionally, let $\alpha \in [0, 2]$ represent the alignment between $\boldsymbol{G}$ and the low-rank subspaces $\mathcal{R}(\boldsymbol{A})$ and $\mathcal{C}(\boldsymbol{B})$ such that: $\|\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}\|_F^2 + \|\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})}\boldsymbol{G}\|_F^2 = \alpha\|\boldsymbol{G}\|_F^2$. Under these assumptions, we derive the following theorem regarding the linear convergence rates of LoRA and SDS-LoRA:

Theorem 3.2. *Assume that the μ-PL condition holds near the pretrained weight $\boldsymbol{W}_0$, that the loss $\mathcal{L}$ is β-smooth, and that the subspaces of $\boldsymbol{A}$ and $\boldsymbol{B}$ capture a fraction α of the energy of $\boldsymbol{G}$. Then, the*

Algorithm 1: Training Procedure of SDS-LoRA

Input: Pretrained weight $\boldsymbol{W}_0$, trainset $\mathcal{D} = \{\mathcal{B}_1, \ldots \mathcal{B}_{T_{\text{total}}}\}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ s.t. $\boldsymbol{BA} = \boldsymbol{0}$, and warm-up iterations T.

`// Warm-Up Stage`

for $t = 1$ **to** T **do**

  $\Delta \boldsymbol{W}_{\text{LoRA}} \leftarrow s\boldsymbol{BA}$

  Forward pass on $\mathcal{B}_t$ using $\boldsymbol{W}_{\text{eff}} = \boldsymbol{W}_0 + \Delta \boldsymbol{W}_{\text{LoRA}}$

  Backward pass and update $\boldsymbol{A}$ and $\boldsymbol{B}$

  if $t = T$ **then**

    Compute the rank-r truncated SVD of $\Delta \boldsymbol{W}_{\text{LoRA}}$: $\Delta \boldsymbol{W}_{\text{LoRA}} = \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r^\top$

    $\boldsymbol{A} \leftarrow \frac{1}{2s} \boldsymbol{\Sigma}_r \boldsymbol{V}_r^\top$, $\boldsymbol{B} \leftarrow \frac{1}{2s} \boldsymbol{U}_r \boldsymbol{\Sigma}_r$, $\boldsymbol{Q_A} \leftarrow \boldsymbol{V}_r$, $\boldsymbol{Q_B} \leftarrow \boldsymbol{U}_r$ `// Equation 8`

    Clear all optimizer states

`// Main Training Stage`

for $t = 1$ **to** T_{total} **do**

  $\Delta \boldsymbol{W}_{\text{SDS-LoRA}} \leftarrow s(\boldsymbol{Q_B A} + \boldsymbol{B Q_A^\top})$ `// Equation 5`

  Forward pass on $\mathcal{B}_t$ using $\boldsymbol{W}_{\text{eff}} = \boldsymbol{W}_0 + \Delta \boldsymbol{W}_{\text{SDS-LoRA}}$

  Backward pass and update $\boldsymbol{A}$ and $\boldsymbol{B}$

  if $t \bmod \left\lceil \frac{kt}{T_{total}} \right\rceil = 0$ **then**

    Compute the QR decomposition of $\boldsymbol{A}^\top$ and $\boldsymbol{B}$: $\boldsymbol{A}^\top = \boldsymbol{Q'_A R'_A}$ and $\boldsymbol{B} = \boldsymbol{Q'_B R'_B}$

    $\boldsymbol{Q_A} \leftarrow \boldsymbol{Q'_A}$, $\boldsymbol{Q_B} \leftarrow \boldsymbol{Q'_B}$

return $\boldsymbol{A}, \boldsymbol{B}$

Table 1: **Results on commonsense reasoning tasks.**

Model	Method	Rank	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-c	ARC-e	OBQA	Avg.
Gemma-2B	Full FT.	-	66.96±0.21	80.31±0.12	75.85±0.47	85.48±0.43	75.61±0.67	65.16±0.70	80.32±0.38	76.27±0.34	75.75
	LoRA	8	63.94±0.04	76.46±0.28	70.62±0.36	44.50±0.48	65.48±0.74	57.20±0.28	75.06±0.56	65.60±1.41	64.86
	rsLoRA	8	64.58±0.07	77.68±0.56	72.65±0.60	55.14±1.73	69.48±0.15	58.70±0.00	76.19±0.38	69.40±0.57	67.98
	LoRA+	8	63.35±0.00	76.82±0.03	72.00±0.36	77.18±1.01	70.90±1.49	58.47±0.44	76.06±0.02	70.20±0.28	70.62
	PiSSA	8	64.94±0.20	77.49±0.13	72.79±0.31	59.89±3.06	69.27±0.04	58.22±0.52	76.18±0.12	71.33±0.19	68.77
	DoRA	8	64.72±0.16	77.57±0.41	72.77±0.36	56.56±0.61	69.82±0.41	59.02±0.56	75.94±0.08	69.73±0.47	68.27
	SDS-LoRA	8	**65.86**±0.44	**79.10**±0.66	**74.48**±0.63	**80.31**±0.45	**73.08**±0.55	**62.08**±0.78	**78.52**±0.36	**73.19**±0.49	**73.33**
	LoRA	32	64.24±0.10	76.22±0.15	70.71±0.05	45.05±1.40	65.98±0.11	57.51±0.24	74.96±0.18	65.13±0.47	64.98
	rsLoRA	32	65.63±0.04	78.89±0.62	73.97±0.17	77.69±0.00	71.03±0.45	61.26±0.72	78.55±0.08	72.80±1.41	72.48
	LoRA+	32	64.62±0.14	78.32±0.10	73.18±0.65	81.73±0.10	71.48±0.45	61.59±0.20	76.91±0.30	73.20±0.28	72.63
	PiSSA	32	65.74±0.32	77.98±0.05	74.14±0.34	72.90±1.05	70.96±0.33	59.47±0.97	76.91±0.28	74.40±0.28	71.56
	DoRA	32	65.30±0.07	78.89±0.77	74.19±0.10	74.17±1.16	71.56±0.19	61.06±0.68	78.56±0.20	72.60±1.41	72.04
	SDS-LoRA	32	**66.34**±0.35	**80.78**±0.53	**74.86**±0.58	**84.93**±0.52	**74.01**±0.64	**62.84**±0.85	**79.34**±0.44	**75.29**±0.82	**74.80**
LLaMA3-8B	Full FT.	-	75.24±0.21	89.70±0.22	81.51±0.24	96.11±0.15	87.66±0.17	81.23±0.54	92.07±0.19	88.02±0.31	86.44
	LoRA	8	72.98±0.06	87.60±0.31	79.67±0.39	94.35±0.14	83.61±0.15	78.95±0.44	90.14±0.02	83.87±1.04	83.89
	rsLoRA	8	73.40±0.19	88.23±0.26	80.26±0.41	95.19±0.03	84.98±0.19	79.21±0.32	90.70±0.12	84.80±0.28	84.60
	LoRA+	8	73.07±0.61	88.03±0.44	80.22±0.29	94.75±0.01	85.00±0.07	78.95±0.36	90.16±0.06	85.20±0.00	84.42
	PiSSA	8	73.68±0.01	88.16±0.10	80.37±0.19	95.20±0.05	85.82±0.19	**80.12**±0.60	90.36±0.06	85.67±0.66	84.92
	DoRA	8	73.20±0.01	87.85±0.10	80.21±0.27	95.22±0.00	84.37±0.45	79.66±0.16	90.53±0.12	85.53±0.19	84.57
	SDS-LoRA	8	**73.91**±0.41	**88.77**±0.80	**80.91**±0.40	**96.00**±0.57	**87.08**±0.52	79.90±0.45	**90.80**±0.74	**86.02**±0.48	**85.42**
	LoRA	32	73.06±0.13	87.45±0.05	79.68±0.00	94.50±0.08	83.32±0.15	79.58±0.52	90.31±0.02	84.07±0.75	84.00
	rsLoRA	32	72.56±0.89	88.47±0.08	80.74±0.39	91.80±5.36	85.61±0.52	79.69±0.60	90.90±0.08	85.53±1.04	84.41
	LoRA+	32	73.43±0.30	88.25±0.10	80.03±0.05	94.99±0.01	85.87±0.41	79.92±0.20	90.05±0.32	85.73±1.04	84.78
	PiSSA	32	74.46±0.52	89.03±0.44	80.79±0.41	95.48±0.02	86.98±0.11	80.12±0.36	90.75±0.14	**86.33**±0.66	85.49
	DoRA	32	73.86±0.23	88.72±0.33	80.98±0.27	95.67±0.05	85.85±0.41	79.78±0.48	91.08±0.00	85.87±0.09	85.23
	SDS-LoRA	32	**74.69**±0.49	**89.43**±0.66	**81.05**±0.58	**96.21**±0.75	**88.08**±0.55	**81.03**±0.77	**92.01**±0.34	86.18±0.78	**86.08**

linear convergence rates of standard LoRA and SDS-LoRA are given by:

$$\mathcal{L}_{t+1} - \mathcal{L}^* \leq \begin{cases} (1 - \frac{\mu\alpha}{2\beta\kappa^4})(\mathcal{L}_t - \mathcal{L}^*) & \text{(LoRA)} \\ (1 - \frac{\mu\alpha}{2\beta})(\mathcal{L}_t - \mathcal{L}^*) & \text{(SDS-LoRA)} \end{cases} \tag{9}$$

where $\mathcal{L}^*$ denotes the minimal loss and $\kappa = \frac{\max(\sigma_{\max}(\boldsymbol{A}), \sigma_{\max}(\boldsymbol{B}))}{\min(\sigma_{\min}(\boldsymbol{A}), \sigma_{\min}(\boldsymbol{B}))}$.

See Section B for the proof and details. Theorem 3.2 suggests that as the singular values of $\boldsymbol{A}$ and $\boldsymbol{B}$ become more skewed, the condition number κ increases, thereby degrading the convergence rate of LoRA. In contrast, since singular values do not influence the backward pass in SDS-LoRA, its convergence rate is independent of κ. Our theoretical analysis further highlights the significance of addressing anisotropic gradient scaling in LoRA.

Table 2: **Results on natural language generation tasks.**

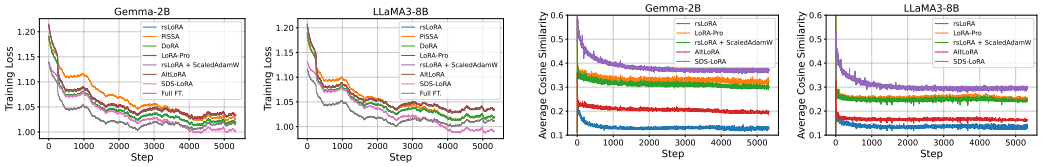
Method	Rank	Gemma-2B			LLaMA3-8B		
		MATH	GSM8K	HumanEval	MATH	GSM8K	HumanEval
Full FT.	-	19.17±0.22	56.23±0.23	33.35±0.65	26.47±0.41	77.04±0.18	49.11±0.44
LoRA	8	16.12±0.28	45.59±0.95	27.24±0.57	23.68±0.28	73.44±0.22	42.53±1.80
rsLoRA	8	17.10±0.24	49.10±1.44	29.27±1.79	24.52±0.09	75.71±0.23	42.23±1.63
LoRA+	8	17.09±0.46	50.30±0.81	27.64±0.76	24.65±0.63	75.27±0.59	45.27±2.84
PiSSA	8	16.44±0.38	50.97±0.74	28.86±0.58	24.43±0.23	74.96±1.01	46.19±2.64
DoRA	8	17.19±0.28	50.61±0.85	28.35±1.52	24.63±0.21	75.41±1.32	43.29±1.14
SDS-LoRA	8	**17.62**±0.66	**52.01**±0.72	**31.24**±1.43	**25.81**±0.53	**76.42**±0.90	**46.95**±1.56
LoRA	32	16.31±0.13	45.67±0.91	28.66±0.50	23.66±0.25	73.49±0.59	42.53±1.09
rsLoRA	32	18.09±0.04	51.93±0.53	31.10±1.32	25.97±0.17	76.28±0.96	44.36±1.39
LoRA+	32	17.45±0.39	52.74±0.75	31.30±1.25	25.01±0.11	76.11±0.20	46.34±1.29
PiSSA	32	17.23±0.24	53.38±0.77	32.11±0.76	24.80±0.41	76.39±0.75	47.10±2.38
DoRA	32	18.27±0.14	52.41±0.24	30.79±1.52	25.87±0.42	76.42±0.17	44.97±1.00
SDS-LoRA	32	**18.31**±0.54	**55.21**±0.81	**34.01**±1.50	**26.36**±0.73	**77.13**±0.84	**47.61**±1.67

Table 3: **Results on image classification tasks with ViT-Base.** See Section G for details and more results.

Method	Rank	Cars	CUB200	SUN397
Full FT.	-	84.26±0.19	86.35±0.18	74.76±0.26
LoRA	8	78.66±0.23	85.65±0.05	74.35±0.28
rsLoRA	8	78.48±0.07	85.67±0.41	74.63±0.13
LoRA+	8	79.05±0.25	85.28±0.11	72.87±0.28
PiSSA	8	78.03±0.32	85.17±0.24	72.82±0.10
DoRA	8	78.93±0.30	85.74±0.35	74.46±0.09
SDS-LoRA	8	**80.41**±0.20	**86.07**±0.13	**74.85**±0.13
LoRA	32	81.48±0.20	85.75±0.28	70.82±0.07
rsLoRA	32	80.57±0.22	85.42±0.41	73.92±0.26
LoRA+	32	79.53±0.60	85.68±0.31	67.66±0.06
PiSSA	32	80.43±0.17	84.90±0.28	69.73±0.12
DoRA	32	81.07±0.03	85.67±0.22	74.05±0.21
SDS-LoRA	32	**82.95**±0.25	**85.99**±0.22	**74.89**±0.14

Table 4: **Comparison with related methods.** We reproduce the results of existing methods. We set the rank to 8.

Method	Gemma-2B			LLaMA3-8B		
	MATH	GSM8K	HumanEval	MATH	GSM8K	HumanEval
LoRA-GA	17.02±0.28	48.60±0.74	30.69±0.76	24.74±0.23	75.21±1.01	45.12±1.22
LoRA-Pro	16.52±0.14	44.58±0.55	28.66±0.99	23.90±0.44	72.93±0.17	40.24±0.99
rsLoRA + ScaledAdamW	16.64±0.31	45.56±0.53	26.83±0.99	24.00±0.17	73.39±0.20	43.29±1.15
AltLoRA	16.62±0.28	45.34±0.81	28.66±0.91	23.66±0.68	73.69±0.23	42.07±0.82
SDS-LoRA	**17.62**±0.66	**52.01**±0.72	**31.24**±1.43	**25.81**±0.53	**76.42**±0.90	**46.95**±1.56



(a) Training loss curve

(b) Alignment between $\boldsymbol{G}$ and $\tilde{\boldsymbol{G}}$

Figure 3: **Comparison of loss convergence and gradient approximation quality.** (a) We smooth the loss curve to enhance visual clarity. (b) We report the average cosine similarity between $\boldsymbol{G}$ and $\tilde{\boldsymbol{G}}$ across all layers. Models are fine-tuned using the Commonsense-170K dataset with rank 32.

4 Experiments

4.1 Experimental Setup

Models and Datasets. For commonsense reasoning tasks, we fine-tune on the Commonsense-170K dataset [20] and evaluate on eight standard benchmarks, namely BoolQ [11], PIQA [5], SIQA [42], HellaSwag [57], WinoGrande [41], ARC-c/e [12], and OBQA [38]. For natural language generation tasks, we select 100K examples from the MetaMathQA [55] and Code-Feedback [62] datasets for fine-tuning. To assess performance, we use MATH [18] and GSM8K [13] benchmarks for the models fine-tuned on MetaMathQA, and HumanEval [9] benchmark for the models fine-tuned on Code-Feedback. For the details on image classification tasks, see Section G.

Implementation Details. For optimization, we use the AdamW optimizer [36] with standard settings. We set $\beta_1 = 0.9$, $\beta_2 = 0.999$, and apply zero weight decay to the optimizer. For SDS-LoRA, we set the warm-up iteration T to 10 and the update scheduling hyperparameter k to 5. For the scaling factor s, we fix it to 2 for LoRA and $\frac{\alpha}{\sqrt{r}}$ for the other methods, following rsLoRA [23], where α is set to 4. We compare our method with the original LoRA and its recent variants, including rsLoRA [23], LoRA+ [17] with a learning rate ratio of 4, PiSSA [37], and DoRA [35]. We reproduce the results of these methods under the same setting. We report the mean and standard deviation over three trials. All experiments are conducted on NVIDIA H200 GPUs. More details are provided in Table 16.

4.2 Commonsense Reasoning

Table 1 shows the experimental results on commonsense reasoning tasks. SDS-LoRA outperforms LoRA and its variants in nearly all configurations, leading to the highest adaptation performance on average across all models and ranks. Specifically, SDS-LoRA yields improvements of approximately 9 and 2 percentage points over LoRA on Gemma-2B and LLaMA3-8B, respectively, and surpasses recent variants including PiSSA and DoRA by 3–5 and 0.5–1 percentage points on Gemma-2B and LLaMA3-8B, respectively. In particular, Gemma-2B with SDS-LoRA shows a substantial improvement over the others on the challenging HellaSwag benchmark, achieving about a 40 percentage point improvement over LoRA. Compared to full fine-tuning, SDS-LoRA with rank-32 achieves minimal performance gaps, demonstrating the effectiveness of SDS-LoRA.

4.3 Natural Language Generation

Table 2 shows the experimental results on the natural language generation tasks. SDS-LoRA achieves the best adaptation performance on all benchmarks across all models and ranks. Specifically, SDS-LoRA achieves improvements of approximately 1–2, 1–10, and 2–6 percentage points over existing methods on MATH, GSM8K, and HumanEval, respectively. Consistent with the commonsense reasoning results, SDS-LoRA demonstrates substantial performance gains with Gemma-2B, especially in challenging adaptation contexts. Furthermore, SDS-LoRA with rank 32 achieves results comparable to or exceeding those of full fine-tuning. These results demonstrate that SDS-LoRA significantly improves LoRA's performance.

4.4 Image Classification

In addition to natural language processing tasks, we also provide results for image classification tasks. The results in Table 3 demonstrate that our method improves the performance of LoRA substantially, surpassing that of the other methods by significant margins. In particular, we observe that SDS-LoRA achieves about 2-3 percentage points higher accuracy on the Cars dataset, resulting in the smallest performance gap compared to the full fine-tuning case. These results demonstrate the effectiveness of SDS-LoRA in the vision domain. See Section G for experimental details and more results.

4.5 Comparison with Related Methods

We compare SDS-LoRA with existing methods that view LoRA from the low-rank gradient approximation perspective, including LoRA-GA [49], LoRA-Pro [50], ScaledAdamW [58], and Alt-LoRA [56]. We reproduce the results of these methods with the same experimental setup as detailed in Table 16. For a fair comparison, we implement LoRA-Pro without tracking the momentum of full fine-tuning gradients. Table 4 shows that SDS-LoRA outperforms existing methods by a substantial margin. We attribute these results to the fact that while both SDS-LoRA and the previous methods optimize the update rule in the effective weight space, the previous ones fail to account for anisotropic scaling in the LoRA matrices' gradient, leaving them vulnerable to its adverse effects. We further compare SDS-LoRA with LoRA-RITE [54] and RefLoRA [60] in Section H, where we also provide further discussion.

4.6 Loss Convergence and Gradient Approximation Quality

Figure 3a shows the training loss curves for full fine-tuning, SDS-LoRA, and existing methods. The results demonstrate that SDS-LoRA achieves significantly better loss convergence than both standard LoRA (i.e., rsLoRA) and related methods. Notably, our method achieves a lower final training loss than full fine-tuning. Figure 3b illustrates the average cosine similarity between the full gradient $\boldsymbol{G}$ and its low-rank approximation $\tilde{\boldsymbol{G}}$ across all layers. We observe that SDS-LoRA exhibits higher alignment throughout training than the existing methods. As discussed in Section H, we attribute the performance gap between SDS-LoRA and the compared methods (i.e., preconditioning-based methods) to the fact that our method overcomes anisotropic gradient scaling in the LoRA matrices' gradients, whereas the preconditioning-based methods do not. This highlights the significance of addressing anisotropic gradient scaling. Overall, these results support our theoretical analysis of both the quality of the gradient approximation (Theorem 3.1) and the convergence rate (Theorem 3.2).

Table 5: **Ablation studies.** 'Full RC of $\Delta \boldsymbol{W}$' indicates the full representational capacity of $\Delta \boldsymbol{W}$. Models are fine-tuned with rank 32.

Formulation of $\Delta \boldsymbol{W}$	Anisotropic Gradient Scaling	Full RC of $\Delta \boldsymbol{W}$	Gemma-2B			LLaMA3-8B		
			MATH	GSM8K	HumanEval	MATH	GSM8K	HumanEval
Full FT.	-	-	19.17$_{\pm 0.22}$	56.23$_{\pm 0.23}$	33.35$_{\pm 0.65}$	26.47$_{\pm 0.41}$	77.04$_{\pm 0.18}$	49.11$_{\pm 0.44}$
$s\boldsymbol{BA}$	✔	✔	18.09$_{\pm 0.04}$	51.93$_{\pm 0.53}$	31.10$_{\pm 1.32}$	25.97$_{\pm 0.17}$	76.28$_{\pm 0.96}$	44.36$_{\pm 1.39}$
$s\boldsymbol{Q_B Q_A^\top}$ (trainable $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$)	✘	✘	17.22$_{\pm 0.24}$	51.18$_{\pm 0.29}$	31.30$_{\pm 0.99}$	24.14$_{\pm 0.27}$	75.68$_{\pm 0.35}$	45.04$_{\pm 1.02}$
$s(\boldsymbol{Q_B A} + \boldsymbol{B Q_A^\top})$ **(Ours)**	✘	✔	**18.31**$_{\pm 0.54}$	**55.21**$_{\pm 0.81}$	**34.01**$_{\pm 1.50}$	**26.36**$_{\pm 0.73}$	**77.13**$_{\pm 0.84}$	**47.61**$_{\pm 1.67}$

Table 6: **Analysis of training overhead.** We train the LLaMA3-8B model on a single NVIDIA H200 GPU for one epoch. For SDS-LoRA, we report the training time for the warm-up (first term) and main training phases (second term) separately.

Method	MetaMathQA		Code-Feedback	
	Training Time (hours)	GPU-Memory (GB)	Training Time (hours)	GPU-Memory (GB)
LoRA	1.61	128.54	1.72	128.80
SDS-LoRA	0.01 + 1.67 (**+4.35%**)	129.16 (**+0.48%**)	0.01 + 1.80 (**+5.23%**)	129.77 (**+0.75%**)

4.7 Ablation Studies

To justify the design choice of SDS-LoRA $\Delta \boldsymbol{W}_{\text{SDS-LoRA}} = s(\boldsymbol{Q_B A} + \boldsymbol{B Q_A^\top})$, we compare it against the LoRA formulation and a variant where weight updates are defined as $\Delta \boldsymbol{W} = s\boldsymbol{Q_B Q_A^\top}$, with both $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$ being learnable. Specifically, for this variant, we compute the QR decompositions of $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$ after their optimization step and replace them with new ones to ensure that their columns remain orthonormal. Although this approach prevents anisotropic gradient scaling, its representational capacity is severely limited because the non-zero singular values of the weight updates are forced to be uniform. In contrast, SDS-LoRA eliminates anisotropic gradient scaling without compromising the representational capacity of weight updates. As shown in Table 5, SDS-LoRA significantly outperforms both standard LoRA and the variant, each of which satisfies only one of the two desirable properties. These results suggest that addressing anisotropic gradient scaling in LoRA is not straightforward due to the dual role of singular values in the forward and backward passes. Additional ablation studies, on orthogonalization-based variants (Section I) and on the warm-up stage (Section D), are provided in the appendix.

4.8 Training Overhead

We measure the training time and GPU memory of SDS-LoRA relative to LoRA. Table 6 shows that SDS-LoRA introduces approximately 5% overhead in time and less than 1% overhead in GPU-memory. These results further highlight the efficiency of SDS-LoRA.

5 Conclusion

In this paper, we provide a novel perspective on the limitations of Low-Rank Adaptation (LoRA). Our analysis reveals that LoRA suffers from anisotropic gradient scaling induced by singular values, which distorts the rich information contained in the full fine-tuning gradient and causes LoRA to deviate from full fine-tuning. To overcome these limitations, we propose a new parameterization for low-rank weight updates that structurally decouples the singular values from the backward pass. Our convergence analysis confirms that the proposed method eliminates the dependence of the convergence rate on the condition number of the LoRA matrices. Experimental results across various benchmarks and pretrained models show that our method significantly accelerates loss convergence and narrows the gap to full fine-tuning, achieving superior adaptation performance.

Limitations. In the proposed method, the effective gradient with respect to $\boldsymbol{W}_{\text{eff}}$ deviates slightly from the worst-case-optimal form, as represented by $\boldsymbol{E}$ in Equation 7. This deviation arises from the updates to $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$. Although we demonstrate empirically in Section E that the contribution of this term is small (see also Table 10), we do not provide a formal justification. For theoretical clarity, we assume this gap is negligible in the convergence analysis (Assumption B.3). Also, the convergence analysis assumes idealized SGD training scenarios, which differ from the AdamW setting used in our experiments.

Acknowledgments and Disclosure of Funding

This work was supported in part by the IITP grants [No. RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University), No. RS-2025-02303870, No. RS-2025-25422680, No.2022-0-00156] funded by the Korean government (MSIT) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-24533064). We also thank Nota Inc. for supporting the revision of this paper and providing travel support for its presentation at NeurIPS 2026.

References

[1] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *ACL*, 2021.

[2] Paul Albert, Frederic Z. Zhang, Hemanth Saratchandran, Cristian Rodriguez-Opazo, Anton van den Hengel, and Ehsan Abbasnejad. Randlora: Full-rank parameter-efficient fine-tuning of large models. In *ICLR*, 2025.

[3] Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee. Meta-learning with adaptive hyperparameters. In *NeurIPS*, 2020.

[4] Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee. Learning to learn task-adaptive hyperparameters for few-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[5] Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*, 2020.

[6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

[8] Kerim Büyükakyüz. Olora: Orthonormal low-rank adaptation of large language models. *arXiv preprint arXiv:2406.01775*, 2024.

[9] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[10] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

[11] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *NAACL-HLT*, 2019.

[12] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[15] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[16] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. The impact of initialization on lora finetuning dynamics. In *NeurIPS*, 2024.

[17] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. In *ICML*, 2024.

[18] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *NeurIPS*, 2021.

[19] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

[20] Zhiqiang Hu, Yihuai Lan, Lei Wang, Wanyu Xu, Ee-Peng Lim, Roy Ka-Wei Lee, Lidong Bing, and Soujanya Poria. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*, 2023.

[21] Qiushi Huang, Tom Ko, Zhan Zhuang, Lilian Tang, and Yu Zhang. Hira: Parameter-efficient hadamard high-rank adaptation for large language models. In *ICLR*, 2025.

[22] Jinwook Jung, Taegyu Kim, Kumju Jo, and Sungyong Baik. Learning to recover task experts from a multi-task merged model. In *ECCV*, 2026.

[23] Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*, 2023.

[24] Wenjun Ke, Jiahao Wang, Peng Wang, Jiajun Liu, Dong Nie, Guozheng Li, and Yining Li. Unveiling lora intrinsic ranks via salience analysis. In *NeurIPS*, 2024.

[25] Soroush Abbasi Koohpayegani, KL Navaneet, Parsa Nooralinejad, Soheil Kolouri, and Hamed Pirsiavash. Nola: Compressing lora using linear combination of random basis. In *ICLR*, 2024.

[26] Dawid J. Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. Vera: Vector-based random matrix adaptation. In *ICLR*, 2024.

[27] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013.

[28] Yeoreum Lee, Jinwook Jung, and Sungyong Baik. Mitigating parameter interference in model merging via sharpness-aware fine-tuning. In *ICLR*, 2025.

[29] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *ICLR*, 2018.

[30] Shiwei Li, Xiandi Luo, Xing Tang, Haozhao Wang, Hao Chen, Weihong Luo, Yuhua Li, Xiuqiang He, and Ruixuan Li. Beyond zero initialization: Investigating the impact of non-zero initialization on lora fine-tuning dynamics. In *ICML*, 2025.

[31] Zhizhong Li, Sina Sajadmanesh, Jingtao Li, and Lingjuan Lyu. Stella: Subspace learning in low-rank adaptation using stiefel manifold. In *NeurIPS*, 2025.

[32] Vladislav Lialin, Namrata Shivagunde, Sherin Muckatira, and Anna Rumshisky. Relora: High-rank training through low-rank updates. In *ICLR*, 2024.

[33] Vijay Lingam, Atula Tejaswi, Aditya Vavre, Aneesh Shetty, Gautham Krishna Gudur, Joydeep Ghosh, Alex Dimakis, Eunsol Choi, Aleksandar Bojchevski, and Sujay Sanghavi. Svft: Parameter-efficient fine-tuning with singular vectors. In *NeurIPS*, 2024.

[34] Kai Lion, Liang Zhang, Bingcong Li, and Niao He. Polar: Polar-decomposed low-rank adapter representation. In *NeurIPS*, 2025.

[35] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *ICML*, 2024.

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[37] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. In *NeurIPS*, 2024.

[38] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

[39] Yurii Nesterov. *Lectures on Convex Optimization*. Springer, Cham, Switzerland, 2018.

[40] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *Technical report, OpenAI*, 2018.

[41] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *AAAI*, 2020.

[42] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *EMNLP-IJCNLP*, 2019.

[43] Chongjie Si, Zhiyi Shi, Shifan Zhang, Xiaokang Yang, Hanspeter Pfister, and Wei Shen. Unleashing the power of task-specific directions in parameter efficient fine-tuning. In *ICLR*, 2025.

[44] Jungyong Son, Jinwook Jung, and Sungyong Baik. Training-free task classification for multi-task model merging. In *ECCV*, 2026.

[45] Nurbek Tastan, Stefanos Laskaridis, Martin Takac, Karthik Nandakumar, and Samuel Horvath. Loft: Low-rank adaptation that behaves like full fine-tuning. In *ICLR*, 2026.

[46] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[48] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.

[49] Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. In *NeurIPS*, 2024.

[50] Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Lora-pro: Are low-rank adapters properly optimized? In *ICLR*, 2025.

[51] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[52] Ziqing Xu, Hancheng Min, Salma Tarmoun, Enrique Mallada, and Rene Vidal. A local polyak-lojasiewicz and descent lemma of gradient descent for overparametrized linear models. *TMLR*, 2025.

[53] Jiahui Yang, Yongjia Ma, Donglin Di, Hao Li, Wei Chen, Yan Xie, Jianxun Cui, Xun Yang, and Wangmeng Zuo. Qr-lora: Efficient and disentangled fine-tuning via qr decomposition for customized generation. In *ICCV*, 2025.

[54] Jui-Nan Yen, Si Si, Zhao Meng, Felix Yu, Sai Surya Duvvuri, Inderjit S. Dhillon, Cho-Jui Hsieh, and Sanjiv Kumar. Lora done rite: Robust invariant transformation equilibration for lora optimization. In *ICLR*, 2025.

[55] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *ICLR*, 2024.

[56] Xin Yu, Yujia Wang, Jinghui Chen, and Lingzhou Xue. Altlora: Towards better gradient approximation in low-rank adaptation with alternating projections. In *NeurIPS*, 2025.

[57] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *ACL*, 2019.

[58] Fangzhao Zhang and Mert Pilanci. Riemannian preconditioned lora for fine-tuning foundation models. In *ICML*, 2024.

[59] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. In *ICLR*, 2023.

[60] Yilang Zhang, Bingcong Li, and Georgios B. Giannakis. Reflora: Refactored low-rank adaptation for efficient fine-tuning of large models. In *NeurIPS*, 2025.

[61] Yuanhe Zhang, Fanghui Liu, and Yudong Chen. Lora-one: One-step full gradient could suffice for fine-tuning large language models, provably and efficiently. In *ICML*, 2025.

[62] Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. Opencodeinterpreter: Integrating code generation with execution and refinement. *arXiv preprint arXiv:2402.14658*, 2024.

The contents of the Appendix are as follows:

- In Section A, we provide the proof of Theorem 3.1.
- In Section B, we provide the details and proof of Theorem 3.2.
- In Section C, we provide ablation studies on the update scheduling of $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$ in our method.
- In Section D, we provide ablation studies on the warm-up stage and on the update of $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$.
- In Section E, we provide empirical evidence showing that the subspace of the LoRA matrices changes slowly in SDS-LoRA.
- In Section F, we show that the singular values of the LoRA matrices exhibit a highly skewed spectrum.
- In Section G, we provide the details and experimental results on image classification tasks.
- In Section H, we provide a detailed discussion on the difference between SDS-LoRA and preconditioning-based methods.
- In Section I, we compare SDS-LoRA with orthogonalization-based LoRA variants.
- In Table 16, we provide additional implementation details.

A Proof of Theorem 3.1

Proof. Recall $\boldsymbol{M_A} = \boldsymbol{V_A}^\top \boldsymbol{G}^\top \boldsymbol{G} \boldsymbol{V_A}$, $\boldsymbol{M_B} = \boldsymbol{U_B}^\top \boldsymbol{G} \boldsymbol{G}^\top \boldsymbol{U_B}$, $J = \mathrm{Tr}(\boldsymbol{M_A}\boldsymbol{\Sigma_A}^2) + \mathrm{Tr}(\boldsymbol{M_B}\boldsymbol{\Sigma_B}^2)$, and $J^\star(\boldsymbol{G}) = E_{\boldsymbol{A}}\lambda_{\max}(\boldsymbol{M_A}) + E_{\boldsymbol{B}}\lambda_{\max}(\boldsymbol{M_B})$, as defined in Section 3.2. We first verify that $\langle \boldsymbol{G}, \tilde{\boldsymbol{G}} \rangle_F = s^2 J$. Using $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle_F = \mathrm{Tr}(\boldsymbol{X}^\top \boldsymbol{Y})$ and the cyclic property of the trace, the two terms of $\tilde{\boldsymbol{G}}$ in Equation 3 give

$$\begin{aligned} \langle \boldsymbol{G}, \boldsymbol{G}\boldsymbol{V_A}\boldsymbol{\Sigma_A}^2\boldsymbol{V_A}^\top \rangle_F &= \mathrm{Tr}(\boldsymbol{G}\boldsymbol{V_A}\boldsymbol{\Sigma_A}^2\boldsymbol{V_A}^\top\boldsymbol{G}^\top) = \mathrm{Tr}(\boldsymbol{V_A}^\top\boldsymbol{G}^\top\boldsymbol{G}\boldsymbol{V_A}\boldsymbol{\Sigma_A}^2) = \mathrm{Tr}(\boldsymbol{M_A}\boldsymbol{\Sigma_A}^2), \\ \langle \boldsymbol{G}, \boldsymbol{U_B}\boldsymbol{\Sigma_B}^2\boldsymbol{U_B}^\top\boldsymbol{G} \rangle_F &= \mathrm{Tr}(\boldsymbol{G}^\top\boldsymbol{U_B}\boldsymbol{\Sigma_B}^2\boldsymbol{U_B}^\top\boldsymbol{G}) = \mathrm{Tr}(\boldsymbol{U_B}^\top\boldsymbol{G}\boldsymbol{G}^\top\boldsymbol{U_B}\boldsymbol{\Sigma_B}^2) = \mathrm{Tr}(\boldsymbol{M_B}\boldsymbol{\Sigma_B}^2). \end{aligned} \tag{10}$$

Note that $\boldsymbol{M_A}$ and $\boldsymbol{M_B}$ are positive semi-definite by construction, and that the energy constraints are constraints on the spectra, since $\mathrm{Tr}(\boldsymbol{\Sigma_A}^2) = ||\boldsymbol{A}||_F^2 = E_{\boldsymbol{A}}$ and $\mathrm{Tr}(\boldsymbol{\Sigma_B}^2) = ||\boldsymbol{B}||_F^2 = E_{\boldsymbol{B}}$. Throughout, the minimum over $\boldsymbol{G}$ is taken over all $\boldsymbol{G}$ with $J^\star(\boldsymbol{G}) > 0$, that is, over all $\boldsymbol{G}$ that have a non-zero component in the given subspaces; otherwise both J and $J^\star(\boldsymbol{G})$ vanish and their ratio is undefined.

Step 1: the uniform configuration attains a worst-case ratio of $1/r$. For any positive semi-definite $\boldsymbol{X} \in \mathbb{R}^{r\times r}$, its trace equals the sum of its r nonnegative eigenvalues, so $\mathrm{Tr}(\boldsymbol{X}) \geq \lambda_{\max}(\boldsymbol{X})$, with equality if and only if $\mathrm{rank}(\boldsymbol{X}) \leq 1$. At $\boldsymbol{\Sigma_A}^2 = \frac{E_{\boldsymbol{A}}}{r}\boldsymbol{I}_r$ and $\boldsymbol{\Sigma_B}^2 = \frac{E_{\boldsymbol{B}}}{r}\boldsymbol{I}_r$, this gives, for every $\boldsymbol{G}$ with $J^\star(\boldsymbol{G}) > 0$,

$$J = \frac{E_{\boldsymbol{A}}}{r}\mathrm{Tr}(\boldsymbol{M_A}) + \frac{E_{\boldsymbol{B}}}{r}\mathrm{Tr}(\boldsymbol{M_B}) \geq \frac{E_{\boldsymbol{A}}}{r}\lambda_{\max}(\boldsymbol{M_A}) + \frac{E_{\boldsymbol{B}}}{r}\lambda_{\max}(\boldsymbol{M_B}) = \frac{1}{r}J^\star(\boldsymbol{G}),$$

so that $J/J^\star(\boldsymbol{G}) \geq \frac{1}{r}$ for every such $\boldsymbol{G}$, and hence $\min_{\boldsymbol{G}} J/J^\star(\boldsymbol{G}) \geq \frac{1}{r}$ at the uniform configuration. This bound is attained. Let $\hat{\boldsymbol{p}}$ be any unit vector in the row space of $\boldsymbol{A}$ and $\hat{\boldsymbol{u}}$ any unit vector in the column space of $\boldsymbol{B}$, and set $\boldsymbol{G} = \hat{\boldsymbol{u}}\hat{\boldsymbol{p}}^\top$. Then $\boldsymbol{G}^\top\boldsymbol{G} = \hat{\boldsymbol{p}}\hat{\boldsymbol{p}}^\top$ and $\boldsymbol{G}\boldsymbol{G}^\top = \hat{\boldsymbol{u}}\hat{\boldsymbol{u}}^\top$, so that $\boldsymbol{M_A} = (\boldsymbol{V_A}^\top\hat{\boldsymbol{p}})(\boldsymbol{V_A}^\top\hat{\boldsymbol{p}})^\top$ and $\boldsymbol{M_B} = (\boldsymbol{U_B}^\top\hat{\boldsymbol{u}})(\boldsymbol{U_B}^\top\hat{\boldsymbol{u}})^\top$ have rank one for any such $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{u}}$. Consequently $\mathrm{Tr}(\boldsymbol{M_A}) = \lambda_{\max}(\boldsymbol{M_A})$ and $\mathrm{Tr}(\boldsymbol{M_B}) = \lambda_{\max}(\boldsymbol{M_B})$, and the inequality above holds with equality. Hence $\min_{\boldsymbol{G}} J/J^\star(\boldsymbol{G}) = \frac{1}{r}$ exactly at the uniform configuration.

Step 2: no configuration attains a worst-case ratio exceeding $1/r$, and only the uniform configuration attains it. Fix any $\boldsymbol{\Sigma_A}^2$ and $\boldsymbol{\Sigma_B}^2$ satisfying $\mathrm{Tr}(\boldsymbol{\Sigma_A}^2) = E_{\boldsymbol{A}}$ and $\mathrm{Tr}(\boldsymbol{\Sigma_B}^2) = E_{\boldsymbol{B}}$. Let $i_{\min}$ and $j_{\min}$ index their smallest diagonal entries, let $\hat{\boldsymbol{p}}$ be the $i_{\min}$-th column of $\boldsymbol{V_A}$ and $\hat{\boldsymbol{u}}$ the $j_{\min}$-th column of $\boldsymbol{U_B}$, and consider the adversarial gradient $\boldsymbol{G} = \hat{\boldsymbol{u}}\hat{\boldsymbol{p}}^\top$. Since $\boldsymbol{V_A}^\top\hat{\boldsymbol{p}} = \boldsymbol{e}_{i_{\min}}$ and $\boldsymbol{U_B}^\top\hat{\boldsymbol{u}} = \boldsymbol{e}_{j_{\min}}$, the columns of $\boldsymbol{V_A}$ and $\boldsymbol{U_B}$ being orthonormal, the computation in Step 1 gives $\boldsymbol{M_A} = \boldsymbol{e}_{i_{\min}}\boldsymbol{e}_{i_{\min}}^\top$ and $\boldsymbol{M_B} = \boldsymbol{e}_{j_{\min}}\boldsymbol{e}_{j_{\min}}^\top$, so that $\lambda_{\max}(\boldsymbol{M_A}) = \lambda_{\max}(\boldsymbol{M_B}) = 1$ and $J^\star(\boldsymbol{G}) = E_{\boldsymbol{A}} + E_{\boldsymbol{B}}$, while

$$J = \mathrm{Tr}(\boldsymbol{e}_{i_{\min}}\boldsymbol{e}_{i_{\min}}^\top\boldsymbol{\Sigma_A}^2) + \mathrm{Tr}(\boldsymbol{e}_{j_{\min}}\boldsymbol{e}_{j_{\min}}^\top\boldsymbol{\Sigma_B}^2) = \lambda_{\min}(\boldsymbol{\Sigma_A}^2) + \lambda_{\min}(\boldsymbol{\Sigma_B}^2).$$

Thus, for this particular $\boldsymbol{G}$,

$$\frac{J}{J^\star(\boldsymbol{G})} = \frac{\lambda_{\min}(\boldsymbol{\Sigma}_{\boldsymbol{A}}^2) + \lambda_{\min}(\boldsymbol{\Sigma}_{\boldsymbol{B}}^2)}{E_{\boldsymbol{A}} + E_{\boldsymbol{B}}}. \tag{11}$$

The minimum eigenvalue of an $r \times r$ positive semi-definite matrix never exceeds its average eigenvalue, so $\lambda_{\min}(\boldsymbol{\Sigma}_{\boldsymbol{A}}^2) \le \frac{E_{\boldsymbol{A}}}{r}$ and $\lambda_{\min}(\boldsymbol{\Sigma}_{\boldsymbol{B}}^2) \le \frac{E_{\boldsymbol{B}}}{r}$, each with equality if and only if the corresponding spectrum is uniform. Hence $J/J^\star(\boldsymbol{G}) \le \frac{1}{r}$ at this $\boldsymbol{G}$, with strict inequality unless both $\boldsymbol{\Sigma}_{\boldsymbol{A}}^2$ and $\boldsymbol{\Sigma}_{\boldsymbol{B}}^2$ are uniform. Since $\min_{\boldsymbol{G}'} J/J^\star(\boldsymbol{G}') \le J/J^\star(\boldsymbol{G})$ for this particular $\boldsymbol{G}$, we conclude that $\min_{\boldsymbol{G}'} J/J^\star(\boldsymbol{G}') \le \frac{1}{r}$ for every configuration, strictly unless both $\boldsymbol{\Sigma}_{\boldsymbol{A}}^2$ and $\boldsymbol{\Sigma}_{\boldsymbol{B}}^2$ are uniform.

Combining Steps 1 and 2, $\max_{\boldsymbol{\Sigma}_{\boldsymbol{A}}^2, \boldsymbol{\Sigma}_{\boldsymbol{B}}^2} \min_{\boldsymbol{G}} J/J^\star(\boldsymbol{G}) = \frac{1}{r}$, uniquely attained at $\boldsymbol{\Sigma}_{\boldsymbol{A}} = \sqrt{E_{\boldsymbol{A}}/r}\boldsymbol{I}_r$ and $\boldsymbol{\Sigma}_{\boldsymbol{B}} = \sqrt{E_{\boldsymbol{B}}/r}\boldsymbol{I}_r$. □

B Details and Proof of Theorem 3.2

In this section, we provide details and proof of Theorem 3.2. We first formalize the assumptions and definitions for the theorem.

Assumption B.1. (**β-smoothness**) The loss function $\mathcal{L}$ is β-smooth with respect to $\boldsymbol{W}_{\text{eff}}$:

$$\mathcal{L}(\boldsymbol{W}_{\text{eff}} + \Delta\boldsymbol{W}_{\text{eff}}) \le \mathcal{L}(\boldsymbol{W}_{\text{eff}}) + \underbrace{\langle \boldsymbol{G}, \Delta\boldsymbol{W}_{\text{eff}} \rangle_F}_{\text{Descent}} + \underbrace{\frac{\beta}{2}||\Delta\boldsymbol{W}_{\text{eff}}||_F^2}_{\text{Penalty}}. \tag{12}$$

Assumption B.2. (**Local PL condition**) The loss function $\mathcal{L}$ satisfies the μ-Polyak–Łojasiewicz (PL) condition near the pretrained weight $\boldsymbol{W}_0$: $||\nabla_{\boldsymbol{W}_{\text{eff}}}\mathcal{L}||_F^2 \ge 2\mu(\mathcal{L}(\boldsymbol{W}_{\text{eff}}) - \mathcal{L}^*)$ for $\|\boldsymbol{W}_{\text{eff}} - \boldsymbol{W}_0\|_F \le \epsilon$, where $\mathcal{L}^*$ denotes the minimal loss. We assume that $\|\boldsymbol{W}_{\text{eff}} - \boldsymbol{W}_0\|_F \le \epsilon$ remains satisfied throughout training.

Assumption B.3. (**Slow subspace change in SDS-LoRA**) In SDS-LoRA, the change in $\boldsymbol{Q}_{\boldsymbol{A}}$ and $\boldsymbol{Q}_{\boldsymbol{B}}$ is negligible after each update of $\boldsymbol{A}$ and $\boldsymbol{B}$ (i.e., $\Delta\boldsymbol{Q}_{\boldsymbol{A}} \approx 0$ and $\Delta\boldsymbol{Q}_{\boldsymbol{B}} \approx 0$).

See Section E for empirical validation of Assumption B.3.

Definition B.4. Let $\alpha \in [0, 2]$ represent the alignment between $\boldsymbol{G}$ and the low-rank subspaces $\mathcal{R}(\boldsymbol{A})$ and $\mathcal{C}(\boldsymbol{B})$ such that

$$||\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}||_F^2 + ||\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})}\boldsymbol{G}||_F^2 = \alpha||\boldsymbol{G}||_F^2. \tag{13}$$

In Equation 12, two terms determine the loss decrease bound: the descent term and the penalty term. We first derive lemmas on these terms.

Lemma B.5. ***(Descent term)*** *For LoRA, the descent term in Equation 12 is bounded as follows:*

$$\langle \boldsymbol{G}, \Delta\boldsymbol{W}_{\text{eff}} \rangle_F \le -\alpha\eta s^2 \sigma_{\min}^2(\boldsymbol{A}, \boldsymbol{B})||\boldsymbol{G}||_F^2 \quad \textit{(in LoRA)}, \tag{14}$$

where $\sigma_{\min}(\boldsymbol{A}, \boldsymbol{B}) = \min(\sigma_{\min}(\boldsymbol{A}), \sigma_{\min}(\boldsymbol{B}))$. For SDS-LoRA, under Assumption B.3, the descent term is given by

$$\langle \boldsymbol{G}, \Delta\boldsymbol{W}_{\text{eff}} \rangle_F \approx -\alpha\eta s^2||\boldsymbol{G}||_F^2 \quad \textit{(in SDS-LoRA)}. \tag{15}$$

Proof. **1. LoRA**

For LoRA, the gradients with respect to $\boldsymbol{A}$ and $\boldsymbol{B}$ are given by: $\nabla_{\boldsymbol{A}}\mathcal{L} = s\boldsymbol{B}^\top\boldsymbol{G}$ and $\nabla_{\boldsymbol{B}}\mathcal{L} = s\boldsymbol{G}\boldsymbol{A}^\top$. Consequently, the effective weight update is derived as: $\Delta\boldsymbol{W}_{\text{eff}} \approx s(\boldsymbol{B}\Delta\boldsymbol{A} + \Delta\boldsymbol{B}\boldsymbol{A}) = s(\boldsymbol{B}(-\eta s\boldsymbol{B}^\top\boldsymbol{G}) + (-\eta s\boldsymbol{G}\boldsymbol{A}^\top)\boldsymbol{A}) = -\eta s^2(\boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{G} + \boldsymbol{G}\boldsymbol{A}^\top\boldsymbol{A})$. Substituting this into the descent term, we obtain

$$\begin{aligned}
\langle \boldsymbol{G}, \Delta\boldsymbol{W}_{\text{eff}} \rangle_F &= \langle \boldsymbol{G}, -\eta s^2(\boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{G} + \boldsymbol{G}\boldsymbol{A}^\top\boldsymbol{A}) \rangle_F \\
&= -\eta s^2(\langle \boldsymbol{G}, \boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{G} \rangle_F + \langle \boldsymbol{G}, \boldsymbol{G}\boldsymbol{A}^\top\boldsymbol{A} \rangle_F) \\
&= -\eta s^2(||\boldsymbol{B}^\top\boldsymbol{G}||_F^2 + ||\boldsymbol{G}\boldsymbol{A}^\top||_F^2).
\end{aligned} \tag{16}$$

We relate these terms to the projections onto the subspaces $\mathcal{C}(\boldsymbol{B})$ and $\mathcal{R}(\boldsymbol{A})$. For any matrix $\boldsymbol{X}$ and $\boldsymbol{B}$, $||\boldsymbol{B}^\top \boldsymbol{X}||_F^2 = ||\boldsymbol{B}^\top \boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})} \boldsymbol{X}||_F^2 \geq \sigma_{\min}^2(\boldsymbol{B}) ||\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})} \boldsymbol{X}||_F^2$. Applying this inequality:

$$\begin{aligned} ||\boldsymbol{B}^\top \boldsymbol{G}||_F^2 &\geq \sigma_{\min}^2(\boldsymbol{B}) ||\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})} \boldsymbol{G}||_F^2, \\ ||\boldsymbol{G}\boldsymbol{A}^\top||_F^2 &\geq \sigma_{\min}^2(\boldsymbol{A}) ||\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}||_F^2. \end{aligned} \tag{17}$$

Thus, using $\sigma_{\min}^2(\boldsymbol{A}, \boldsymbol{B}) = \min(\sigma_{\min}^2(\boldsymbol{A}), \sigma_{\min}^2(\boldsymbol{B}))$ and Equation 13, we obtain

$$\begin{aligned} \langle \boldsymbol{G}, \Delta \boldsymbol{W}_{\text{eff}} \rangle_F &\leq -\eta s^2 \sigma_{\min}^2(\boldsymbol{A}, \boldsymbol{B}) (||\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})} \boldsymbol{G}||_F^2 + ||\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}||_F^2) \\ &= -\alpha \eta s^2 \sigma_{\min}^2(\boldsymbol{A}, \boldsymbol{B}) ||\boldsymbol{G}||_F^2. \end{aligned} \tag{18}$$

2. SDS-LoRA

Under Assumption B.3, the effective weight update is given by: $\Delta \boldsymbol{W}_{\text{eff}} \approx s(\boldsymbol{Q}_{\boldsymbol{B}} \Delta \boldsymbol{A} + \Delta \boldsymbol{B} \boldsymbol{Q}_{\boldsymbol{A}}^\top + \Delta \boldsymbol{Q}_{\boldsymbol{B}} \boldsymbol{A} + \boldsymbol{B} \Delta \boldsymbol{Q}_{\boldsymbol{A}}^\top) \approx -\eta s^2 (\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})} \boldsymbol{G} + \boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})})$. Substituting this into the inner product:

$$\begin{aligned} \langle \boldsymbol{G}, \Delta \boldsymbol{W}_{\text{eff}} \rangle_F &\approx \langle \boldsymbol{G}, -\eta s^2 (\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})} \boldsymbol{G} + \boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}) \rangle_F \\ &= -\eta s^2 (\langle \boldsymbol{G}, \boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})} \boldsymbol{G} \rangle_F + \langle \boldsymbol{G}, \boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} \rangle_F). \end{aligned} \tag{19}$$

For any matrix $\boldsymbol{X}$ and projection $\boldsymbol{P}$, $\langle \boldsymbol{X}, \boldsymbol{P}\boldsymbol{X} \rangle_F = \text{Tr}(\boldsymbol{X}^\top \boldsymbol{P}\boldsymbol{X}) = \text{Tr}(\boldsymbol{X}^\top \boldsymbol{P}^\top \boldsymbol{P}\boldsymbol{X}) = ||\boldsymbol{P}\boldsymbol{X}||_F^2$. Applying this to our terms:

$$\begin{aligned} \langle \boldsymbol{G}, \boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})} \boldsymbol{G} \rangle_F &= ||\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})} \boldsymbol{G}||_F^2, \\ \langle \boldsymbol{G}, \boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} \rangle_F &= \langle \boldsymbol{G}^\top, \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}^\top \boldsymbol{G}^\top \rangle_F = ||\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}||_F^2. \end{aligned} \tag{20}$$

Finally, applying Equation 13, we obtain

$$\begin{aligned} \langle \boldsymbol{G}, \Delta \boldsymbol{W}_{\text{eff}} \rangle_F &= -\eta s^2 (||\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}||_F^2 + ||\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})} \boldsymbol{G}||_F^2) \\ &= -\alpha \eta s^2 ||\boldsymbol{G}||_F^2. \end{aligned} \tag{21}$$

□

Lemma B.6. ***(Penalty term)*** *For LoRA, the bound for the penalty term in Equation 12 is given by*

$$\textit{Penalty}_{\textit{LoRA}} \leq \alpha \beta \eta^2 s^4 \sigma_{\max}^4(\boldsymbol{A}, \boldsymbol{B}) ||\boldsymbol{G}||_F^2 \quad \textit{(in LoRA)}. \tag{22}$$

For SDS-LoRA, the bound for the penalty term is given by

$$\textit{Penalty}_{\textit{SDS-LoRA}} \leq \alpha \beta \eta^2 s^4 ||\boldsymbol{G}||_F^2 \quad \textit{(in SDS-LoRA)}. \tag{23}$$

where $\sigma_{\max}(\boldsymbol{A}, \boldsymbol{B}) = \max(\sigma_{\max}(\boldsymbol{A}), \sigma_{\max}(\boldsymbol{B}))$.

Proof. We bound the penalty term $\frac{\beta}{2} ||\Delta \boldsymbol{W}_{\text{eff}}||_F^2$ by relating it to the updates of $\boldsymbol{A}$ and $\boldsymbol{B}$. Both parameterizations induce a weight change of the common form $\Delta \boldsymbol{W}_{\text{eff}} \approx s(\boldsymbol{X}\Delta \boldsymbol{A} + \Delta \boldsymbol{B}\boldsymbol{Y})$, where $(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{B}, \boldsymbol{A})$ for LoRA, for which the second-order term $s\Delta \boldsymbol{B}\Delta \boldsymbol{A}$ is dropped, and $(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{Q}_{\boldsymbol{B}}, \boldsymbol{Q}_{\boldsymbol{A}}^\top)$ for SDS-LoRA, for which the form is exact under Assumption B.3. The triangle inequality and sub-multiplicative property of norms give us

$$\begin{aligned} ||\Delta \boldsymbol{W}_{\text{eff}}||_F &\approx s||\boldsymbol{X}\Delta \boldsymbol{A} + \Delta \boldsymbol{B}\boldsymbol{Y}||_F \\ &\leq s(||\boldsymbol{X}||_2 ||\Delta \boldsymbol{A}||_F + ||\Delta \boldsymbol{B}||_F ||\boldsymbol{Y}||_2) \\ &\leq s\rho(||\Delta \boldsymbol{A}||_F + ||\Delta \boldsymbol{B}||_F), \end{aligned} \tag{24}$$

where $\rho = \max(||\boldsymbol{X}||_2, ||\boldsymbol{Y}||_2)$, so that $\rho = \sigma_{\max}(\boldsymbol{A}, \boldsymbol{B})$ for LoRA and $\rho = 1$ for SDS-LoRA. Using $(x+y)^2 \leq 2x^2 + 2y^2$, we obtain

$$||\Delta \boldsymbol{W}_{\text{eff}}||_F^2 \leq 2s^2 \rho^2 (||\Delta \boldsymbol{A}||_F^2 + ||\Delta \boldsymbol{B}||_F^2). \tag{25}$$

1. LoRA

We relate the norm of $\Delta \boldsymbol{A} = -\eta s \boldsymbol{B}^\top \boldsymbol{G}$ and $\Delta \boldsymbol{B} = -\eta s \boldsymbol{G}\boldsymbol{A}^\top$ to the subspace projections:

$$\begin{aligned} ||\Delta \boldsymbol{A}||_F^2 &= \eta^2 s^2 ||\boldsymbol{B}^\top \boldsymbol{G}||_F^2 = \eta^2 s^2 ||\boldsymbol{B}^\top \boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})} \boldsymbol{G}||_F^2 \leq \eta^2 s^2 \sigma_{\max}^2(\boldsymbol{B}) ||\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})} \boldsymbol{G}||_F^2, \\ ||\Delta \boldsymbol{B}||_F^2 &= \eta^2 s^2 ||\boldsymbol{G}\boldsymbol{A}^\top||_F^2 = \eta^2 s^2 ||\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} \boldsymbol{A}^\top||_F^2 \leq \eta^2 s^2 \sigma_{\max}^2(\boldsymbol{A}) ||\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}||_F^2. \end{aligned} \tag{26}$$

Substituting these into Equation 25 with $\rho = \sigma_{\max}(\boldsymbol{A}, \boldsymbol{B})$:

$$\begin{aligned} ||\Delta \boldsymbol{W}_{\text{eff}}||_F^2 &\leq 2s^2\sigma_{\max}^2(\boldsymbol{A},\boldsymbol{B})\left(\eta^2 s^2 \sigma_{\max}^2(\boldsymbol{B})||\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})}\boldsymbol{G}||_F^2 + \eta^2 s^2 \sigma_{\max}^2(\boldsymbol{A})||\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}||_F^2\right) \\ &\leq 2\eta^2 s^4 \sigma_{\max}^4(\boldsymbol{A},\boldsymbol{B})\left(||\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})}\boldsymbol{G}||_F^2 + ||\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}||_F^2\right). \end{aligned} \tag{27}$$

Applying Equation 13:

$$||\Delta \boldsymbol{W}_{\text{eff}}||_F^2 \leq 2\alpha\eta^2 s^4 \sigma_{\max}^4(\boldsymbol{A},\boldsymbol{B})||\boldsymbol{G}||_F^2. \tag{28}$$

The penalty is therefore $\frac{\beta}{2}||\Delta \boldsymbol{W}_{\text{eff}}||_F^2 \leq \alpha\beta\eta^2 s^4 \sigma_{\max}^4(\boldsymbol{A},\boldsymbol{B})||\boldsymbol{G}||_F^2$.

2. SDS-LoRA

Since $\boldsymbol{Q}_{\boldsymbol{B}}$ and $\boldsymbol{Q}_{\boldsymbol{A}}^\top$ have orthonormal columns and rows, respectively, their spectral norms are 1. Thus, $\rho = 1$. Using the update rules $\Delta\boldsymbol{A} = -\eta s \boldsymbol{Q}_{\boldsymbol{B}}^\top \boldsymbol{G}$ and $\Delta\boldsymbol{B} = -\eta s \boldsymbol{G}\boldsymbol{Q}_{\boldsymbol{A}}$:

$$\begin{aligned} ||\Delta\boldsymbol{A}||_F^2 &= \eta^2 s^2 ||\boldsymbol{Q}_{\boldsymbol{B}}^\top \boldsymbol{G}||_F^2 = \eta^2 s^2 \operatorname{Tr}(\boldsymbol{G}^\top \boldsymbol{Q}_{\boldsymbol{B}} \boldsymbol{Q}_{\boldsymbol{B}}^\top \boldsymbol{G}) = \eta^2 s^2 ||\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})}\boldsymbol{G}||_F^2. \\ ||\Delta\boldsymbol{B}||_F^2 &= \eta^2 s^2 ||\boldsymbol{G}\boldsymbol{Q}_{\boldsymbol{A}}||_F^2 = \eta^2 s^2 \operatorname{Tr}(\boldsymbol{G}\boldsymbol{Q}_{\boldsymbol{A}} \boldsymbol{Q}_{\boldsymbol{A}}^\top \boldsymbol{G}^\top) = \eta^2 s^2 ||\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}||_F^2. \end{aligned} \tag{29}$$

Substituting these into Equation 25 with $\rho = 1$:

$$||\Delta \boldsymbol{W}_{\text{eff}}||_F^2 \leq 2s^2 \cdot 1^2 \cdot \eta^2 s^2 (||\boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})}\boldsymbol{G}||_F^2 + ||\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}||_F^2). \tag{30}$$

Applying Equation 13:

$$||\Delta \boldsymbol{W}_{\text{eff}}||_F^2 \leq 2\alpha\eta^2 s^4 ||\boldsymbol{G}||_F^2. \tag{31}$$

The penalty is therefore $\frac{\beta}{2}||\Delta \boldsymbol{W}_{\text{eff}}||_F^2 \leq \alpha\beta\eta^2 s^4 ||\boldsymbol{G}||_F^2$. □

Finally, using Lemma B.5 and Lemma B.6, we prove Theorem 3.2.

Proof. Let $\mathcal{L}_t = \mathcal{L}(\boldsymbol{W}_{\text{eff}})$ and $\mathcal{L}_{t+1} = \mathcal{L}(\boldsymbol{W}_{\text{eff}} + \Delta\boldsymbol{W}_{\text{eff}})$.

1. LoRA

Substitute the Descent and Penalty lemmas for LoRA:

$$\mathcal{L}_{t+1} \leq \mathcal{L}_t - \alpha\eta s^2 \sigma_{\min}^2 ||\boldsymbol{G}||_F^2 + \alpha\beta\eta^2 s^4 \sigma_{\max}^4 ||\boldsymbol{G}||_F^2. \tag{32}$$

We set $f(\eta) = -\alpha s^2 \sigma_{\min}^2 \eta + \alpha\beta s^4 \sigma_{\max}^4 \eta^2$. Solving for optimal η^*:

$$f'(\eta^*) = -\alpha s^2 \sigma_{\min}^2 + 2\alpha\beta s^4 \sigma_{\max}^4 \eta = 0 \implies \eta^* = \frac{s^2\sigma_{\min}^2}{2\beta s^4 \sigma_{\max}^4} = \frac{\sigma_{\min}^2}{2\beta s^2 \sigma_{\max}^4}. \tag{33}$$

Substituting η^* back into Equation 32:

$$\begin{aligned} \mathcal{L}_{t+1} &\leq \mathcal{L}_t - \alpha s^2 \sigma_{\min}^2 \left(\frac{\sigma_{\min}^2}{2\beta s^2 \sigma_{\max}^4}\right) ||\boldsymbol{G}||_F^2 + \alpha\beta s^4 \sigma_{\max}^4 \left(\frac{\sigma_{\min}^2}{2\beta s^2 \sigma_{\max}^4}\right)^2 ||\boldsymbol{G}||_F^2 \\ &= \mathcal{L}_t - \frac{\alpha\sigma_{\min}^4}{2\beta\sigma_{\max}^4} ||\boldsymbol{G}||_F^2 + \frac{\alpha\sigma_{\min}^4}{4\beta\sigma_{\max}^4} ||\boldsymbol{G}||_F^2 \\ &= \mathcal{L}_t - \frac{\alpha\sigma_{\min}^4}{4\beta\sigma_{\max}^4} ||\boldsymbol{G}||_F^2. \end{aligned} \tag{34}$$

Recognizing that $\frac{\sigma_{\max}^4}{\sigma_{\min}^4} = \kappa^4$ for $\kappa = \frac{\sigma_{\max}(\boldsymbol{A},\boldsymbol{B})}{\sigma_{\min}(\boldsymbol{A},\boldsymbol{B})}$, we rewrite this as $\mathcal{L}_{t+1} \leq \mathcal{L}_t - \frac{\alpha}{4\beta\kappa^4} ||\boldsymbol{G}||_F^2$. Applying the PL condition:

$$\begin{aligned} \mathcal{L}_{t+1} &\leq \mathcal{L}_t - \frac{\alpha}{4\beta\kappa^4} 2\mu(\mathcal{L}_t - \mathcal{L}^*) \\ \mathcal{L}_{t+1} - \mathcal{L}^* &\leq (1 - \frac{\mu\alpha}{2\beta\kappa^4})(\mathcal{L}_t - \mathcal{L}^*). \end{aligned} \tag{35}$$

2. SDS-LoRA

Substitute the Descent and Penalty lemmas into Equation 12:

$$\mathcal{L}_{t+1} \leq \mathcal{L}_t - \alpha\eta s^2 ||\boldsymbol{G}||_F^2 + \alpha\beta\eta^2 s^4 ||\boldsymbol{G}||_F^2. \tag{36}$$

To find the optimal step size, we define $f(\eta) = -\alpha s^2\eta + \alpha\beta s^4\eta^2$ and solve $f'(\eta^*) = -\alpha s^2 + 2\alpha\beta s^4\eta = 0$. This gives us $\eta^* = \frac{s^2}{2\beta s^4} = \frac{1}{2\beta s^2}$. By substituting η^* back into Equation 36, we obtain:

$$\mathcal{L}_{t+1} \leq \mathcal{L}_t - \frac{\alpha s^2}{2\beta s^2}||\boldsymbol{G}||_F^2 + \frac{\alpha\beta s^4}{4\beta^2 s^4}||\boldsymbol{G}||_F^2 = \mathcal{L}_t - \frac{\alpha}{4\beta}||\boldsymbol{G}||_F^2. \tag{37}$$

Applying the PL condition, we obtain

$$\begin{aligned} \mathcal{L}_{t+1} &\leq \mathcal{L}_t - \frac{\alpha}{4\beta}2\mu(\mathcal{L}_t - \mathcal{L}^*) \\ \mathcal{L}_{t+1} - \mathcal{L}^* &\leq (1 - \frac{\mu\alpha}{2\beta})(\mathcal{L}_t - \mathcal{L}^*). \end{aligned} \tag{38}$$

□

C Ablation Studies on the Update Interval

In this section, we provide ablation studies on the update interval for $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$ in the training procedure of SDS-LoRA described in Algorithm 1. To examine the effectiveness of the proposed interval scheduling (i.e., $t \bmod \left\lceil \frac{kt}{T_{\text{total}}} \right\rceil = 0$), we compare it against a uniform schedule (i.e., $t \bmod k = 0$). Table 7 demonstrates that for both schedules, performance degrades as the update interval increases. This occurs because large update intervals cause the updates of $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$ to significantly disrupt the model, resulting in unstable training. We also observe that frequent updates do not always yield better performance, justifying the use of intermittent updates for efficiency. Furthermore, given the same total number of updates, our scheduling method outperforms the uniform schedule, demonstrating the necessity of frequent updates during the initial training phase.

D Ablation Studies on the Warm-Up Stage and the Update of Orthonormal Bases

Warm-up stage. The warm-up stage serves as a lightweight enabler that defines $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$ for the SDS-LoRA formulation, as described in Equation 8. To examine whether the gain of SDS-LoRA originates from this stage, we ablate both the initialization scheme and the warm-up length in Table 8. Here, 'Random' denotes an initialization in which $\boldsymbol{A}$ and $\boldsymbol{B}$ are randomly initialized and the orthonormal bases are computed from them, after which the resulting $\Delta\boldsymbol{W}$ is subtracted from the pretrained weight so that training still starts from $\boldsymbol{W}_0$. Random initialization degrades performance regardless of the warm-up length, because a randomly defined subspace does not align well with the full fine-tuning gradient, whereas the warm-up stage defines the subspace from actual training examples. A longer warm-up does not help either and slightly degrades performance, which we attribute to the increasing influence of the standard LoRA formulation. Together with the 'SDS-LoRA w/o detachment' variant in Table 15, which shares the same warm-up stage but performs substantially worse, these results indicate that the warm-up stage is a lightweight enabler for the SDS-LoRA parameterization rather than a component to be tuned for performance.

Update of $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$. We also examine what happens when $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$ are never updated after the warm-up stage. Table 9 shows that performance drops substantially in this case, since freezing

Table 7: **Ablation studies on the update interval of $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$.** Experiments are conducted on LLaMA3-8B with rank 32.

Update Condition		Number of Updates ($\times T_{\text{Total}}$)	LLaMA3-8B	
			MATH	GSM8K
$t \bmod k = 0$	$k = 1$	1	26.42±0.76	76.91±0.57
	$k = 2$	0.5	26.24±0.81	77.01±0.73
	$k = 3$	0.33	24.58±0.42	75.50±0.74
$t \bmod \left\lceil \frac{kt}{T_{\text{total}}} \right\rceil = 0$	$k = 3$	0.61	26.53±0.66	77.04±0.72
	$k = 4$	0.52	26.42±0.54	77.16±0.68
	$k = 5$ (Default)	0.46	26.36±0.73	77.13±0.84
	$k = 6$	0.41	25.88±0.44	76.65±0.52
	$k = 7$	0.37	25.03±0.48	76.21±0.78

Table 8: **Ablation studies on the initialization scheme and the warm-up length.** We report the average accuracy on the commonsense reasoning benchmarks with Gemma-2B at rank 32 from a single run.

Initialization	Warm-Up Iterations T	Avg. Accuracy
Random	0	71.24
Random	10	71.16
SVD (ours)	10 (default)	**74.80**
SVD (ours)	50	74.67
SVD (ours)	100	74.54

Table 9: **Ablation study on the update of $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$.** We report the average accuracy on the commonsense reasoning benchmarks with Gemma-2B at rank 32 from a single run.

Update of $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$	Avg. Accuracy
Never after warm-up	64.78
Periodic (ours)	**74.80**

Table 10: **Contribution of the residual term $\boldsymbol{E}$ to the effective gradient.** We divide training into three phases and report the ratio averaged over all layers within each phase. We fine-tune Gemma-2B using the Commonsense-170K dataset with rank 32.

Training progress	Early	Mid	Late
$\|\|\boldsymbol{E}\|\|_F \;/\; \|\|s^2(\boldsymbol{G}\boldsymbol{Q_A}\boldsymbol{Q_A}^\top + \boldsymbol{Q_B}\boldsymbol{Q_B}^\top\boldsymbol{G}) + \boldsymbol{E}\|\|_F$	0.09	0.05	0.03

$\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$ confines the update to a fixed subspace that can no longer effectively capture the full gradient. As reported in Table 6, updating them incurs approximately 5% time and less than 1% memory overhead, which is small relative to the resulting performance gain.

E Subspace Change in SDS-LoRA

In Equation 7, the term $\boldsymbol{E}$, which arises from the change in $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$, causes the effective gradient to deviate from the optimal effective gradient. To assess the contribution of $\boldsymbol{E}$, we examine the rate at which $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$, which represent the subspaces of $\boldsymbol{A}$ and $\boldsymbol{B}$, evolve during training. Specifically, we measure the average cosine similarity between $\boldsymbol{Q_A}$ and $\boldsymbol{Q_A} + \Delta\boldsymbol{Q_A}$ (or $\boldsymbol{Q_B}$ and $\boldsymbol{Q_B} + \Delta\boldsymbol{Q_B}$) across all layers. Figure 4 shows that the cosine similarity remains close to 1 throughout training, except for a few initial iterations. This suggests that the error arising from the subspace change is negligible. Thus, SDS-LoRA nearly achieves the worst-case-optimal gradient quality characterized in Theorem 3.1, providing empirical support for Assumption B.3 in our convergence analysis.

To assess the contribution of $\boldsymbol{E}$ more directly, we additionally measure its relative magnitude within the effective gradient of Equation 7, that is, the ratio $||\boldsymbol{E}||_F / ||\tilde{\boldsymbol{G}}||_F$. Since $\Delta\boldsymbol{W}_{\text{eff}} = -\eta\tilde{\boldsymbol{G}}$, this ratio is exactly the fraction of the effective weight update that originates from the change in $\boldsymbol{Q_A}$ and $\boldsymbol{Q_B}$, as the common factor cancels between the numerator and the denominator. Table 10 shows that the contribution of $\boldsymbol{E}$ stays below 0.1 throughout training and decreases further as training progresses. We acknowledge that this may still be insufficient to strictly justify $\boldsymbol{E} \approx 0$, and SDS-LoRA is therefore best regarded as a practical realization of our theoretical formulation.

F Singular Value Distribution of LoRA Matrices

Our claim regarding the adverse effects of anisotropic gradient scaling, which is induced by the singular values of the LoRA matrices, holds only when the singular values exhibit a highly skewed spectrum. To verify this, we monitor the stable rank of $\boldsymbol{A}$ and $\boldsymbol{B}$ throughout the LoRA training process, defined as $\frac{\|\boldsymbol{A}\|_F^2}{\|\boldsymbol{A}\|_2^2}$ and $\frac{\|\boldsymbol{B}\|_F^2}{\|\boldsymbol{B}\|_2^2}$, where $\|\cdot\|_2$ denotes the spectral norm. Figure 5 demonstrates that the average stable rank for both matrices is significantly lower than the LoRA rank (i.e., 32). This observation confirms that the singular values are indeed highly skewed, thereby validating our claim.

Table 11: **Results on image classification tasks.**

Model	Method	Rank	Cars	CUB200	DTD	Food101	SUN397	Avg.
ViT-Base	Full FT.	-	84.26±0.19	86.35±0.18	80.07±0.32	89.27±0.20	74.76±0.26	82.94
	LoRA	8	78.66±0.23	85.65±0.05	78.81±0.95	88.15±0.07	74.35±0.28	81.12
	rsLoRA		78.48±0.07	85.67±0.41	78.79±0.48	88.13±0.16	74.63±0.13	81.14
	LoRA+		79.05±0.25	85.28±0.11	78.32±0.16	88.27±0.05	72.87±0.28	80.76
	PiSSA		78.03±0.32	85.17±0.24	78.33±0.46	87.65±0.07	72.82±0.10	80.40
	DoRA		78.93±0.30	85.74±0.35	78.46±0.16	88.27±0.17	74.46±0.09	81.17
	SDS-LoRA		**80.41**±0.20	**86.07**±0.13	**79.29**±0.25	**88.31**±0.26	**74.85**±0.13	**81.79**
	LoRA	32	81.48±0.20	85.75±0.28	79.34±0.18	88.37±0.12	70.82±0.07	81.15
	rsLoRA		80.57±0.22	85.42±0.41	79.50±0.09	88.81±0.10	73.92±0.26	81.64
	LoRA+		79.53±0.60	85.68±0.31	78.85±0.39	88.05±0.01	67.66±0.06	79.95
	PiSSA		80.43±0.17	84.90±0.28	78.90±0.35	87.91±0.08	69.73±0.12	80.37
	DoRA		81.07±0.03	85.67±0.22	79.56±0.43	88.83±0.08	74.05±0.21	81.84
	SDS-LoRA		**82.95**±0.25	**85.99**±0.22	**80.10**±0.11	**89.02**±0.16	**74.89**±0.14	**82.59**
ViT-Large	Full FT.	-	88.19±0.13	88.23±0.19	81.71±0.09	90.90±0.17	76.20±0.22	85.05
	LoRA	8	85.07±0.19	87.53±0.20	80.64±0.23	89.96±0.08	76.30±0.11	83.90
	rsLoRA		84.61±0.11	87.75±0.19	80.87±0.32	89.92±0.05	76.45±0.16	83.92
	LoRA+		81.99±0.97	87.66±0.05	80.62±0.55	**90.06**±0.03	75.08±0.14	83.08
	PiSSA		85.59±0.17	87.55±0.19	80.34±0.13	89.60±0.17	75.41±0.20	83.70
	DoRA		84.67±0.08	87.58±0.11	80.73±0.33	90.03±0.04	76.50±0.08	83.90
	SDS-LoRA		**85.73**±0.11	**87.88**±0.22	**81.00**±0.13	89.95±0.16	**76.58**±0.18	**84.23**
	LoRA	32	86.23±0.05	87.94±0.22	81.15±0.32	90.62±0.03	74.59±0.25	84.11
	rsLoRA		85.93±0.30	87.64±0.06	**81.56**±0.07	90.58±0.13	76.22±0.13	84.39
	LoRA+		82.40±1.84	87.92±0.22	80.92±0.18	90.48±0.07	75.19±1.23	83.38
	PiSSA		86.70±0.15	87.65±0.15	80.82±0.26	90.04±0.09	73.84±0.08	83.81
	DoRA		86.10±0.36	87.69±0.11	81.31±0.13	90.64±0.04	76.23±0.20	84.39
	SDS-LoRA		**87.00**±0.25	**88.20**±0.23	81.35±0.10	**90.70**±0.11	**76.46**±0.19	**84.74**

Table 12: **Implementation details on image classification tasks.**

	ViT-Base					ViT-Large				
	Cars	CUB200	DTD	Food101	SUN397	Cars	CUB200	DTD	Food101	SUN397
Epochs	5	7	7	7	5	5	7	7	7	5
Learning Rate	5e-3	2e-3	2e-3	2e-3	5e-3	2.5e-3	1e-3	1e-3	1e-3	2.5e-3
Batch Size	64									
Target Modules	'query', 'value'									

G Experiments on Image Classification Tasks

For experiments on vision benchmarks, we use ViT-Base and ViT-Large [14] pretrained on 224×224 images with a 16×16 patch size. We fine-tune and evaluate them on five datasets, including Cars [27], CUB200 [48], DTD [10], Food101 [6], and SUN397 [51]. We use the same setup as for the natural language processing tasks, except for the epochs, learning rate, and batch size, which are detailed in Table 12. The results in Table 11 show that SDS-LoRA achieves the best average performance across all models and ranks. In particular, we observe that SDS-LoRA significantly reduces the performance gap between LoRA and full fine-tuning. These results suggest that SDS-LoRA generalizes well to vision tasks, highlighting the effectiveness of our method in various domains.

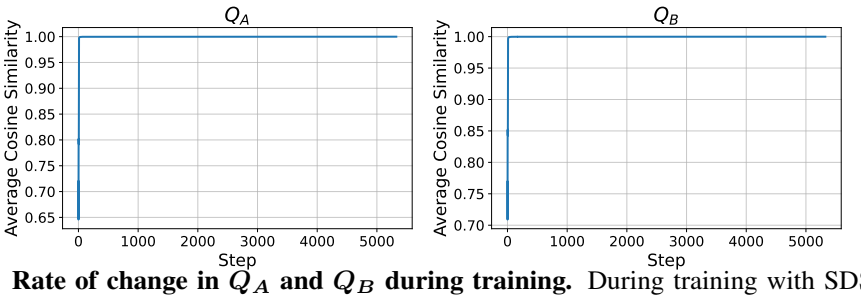


Figure 4: **Rate of change in Q_A and Q_B during training.** During training with SDS-LoRA, we monitor the average cosine similarity between Q_A and $Q_A + \Delta Q_A$ and between Q_B and $Q_B + \Delta Q_B$ across all layers. We fine-tune LLaMA3-8B using the Commonsense-170K dataset with rank 32.

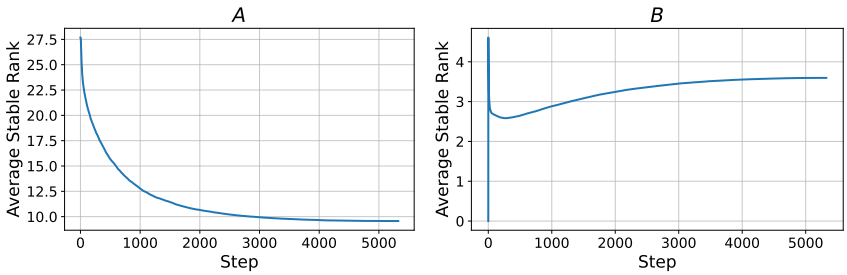


Figure 5: **Stable rank of $\boldsymbol{A}$ and $\boldsymbol{B}$.** We report the average stable rank of $\boldsymbol{A}$ and $\boldsymbol{B}$ across all layers. We fine-tune LLaMA3-8B using the Commonsense-170K dataset with rank 32.

H Comparison to Preconditioning-Based Methods

Several previous works [58, 50, 56, 45] attempt to minimize the gap between the full fine-tuning gradient $\boldsymbol{G}$ and its low-rank approximation $\tilde{\boldsymbol{G}}$ described in Equation 3. Specifically, they modify the original gradient of the LoRA matrices as follows: $\tilde{\nabla}_{\boldsymbol{A}}\mathcal{L} = s(\boldsymbol{B}^\top\boldsymbol{B})^{-1}\boldsymbol{B}^\top\boldsymbol{G}$ and $\tilde{\nabla}_{\boldsymbol{B}}\mathcal{L} = s\boldsymbol{G}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}$ where $(\boldsymbol{B}^\top\boldsymbol{B})^{-1}$ and $(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}$ act as preconditioners. Consequently, the effective gradient of $\boldsymbol{W}_{\text{eff}}$ becomes $\tilde{\boldsymbol{G}} = s^2(\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} + \boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})}\boldsymbol{G})$, attaining the worst-case-optimal form characterized in Theorem 3.1, up to a constant factor that is absorbed into the learning rate.

Although these preconditioning-based methods also achieve optimal gradient approximation, our work differs from theirs in several key aspects. First, our work focuses on addressing the adverse effects of anisotropic scaling induced by singular values, whereas prior works primarily aim to improve the quality of gradient approximation. While both approaches yield the same update rule in the effective weight space $\boldsymbol{W}_{\text{eff}}$, preconditioning-based methods fail to eliminate the influence of singular values in the gradients of the LoRA matrices. Substituting the SVD of the LoRA matrices into $\tilde{\nabla}_{\boldsymbol{A}}\mathcal{L}$, we obtain $\tilde{\nabla}_{\boldsymbol{A}}\mathcal{L} = s\boldsymbol{V}_{\boldsymbol{B}}\boldsymbol{\Sigma}_{\boldsymbol{B}}^{-1}\boldsymbol{U}_{\boldsymbol{B}}^\top\boldsymbol{G}$, indicating that these methods do not address the adverse effects of anisotropic scaling induced by singular values. Second, because preconditioning-based methods modify the original gradients, they do not guarantee the steepest descent direction in the parameter space. The descent lemma suggests that deviations from the true gradient may lead to a weaker bound on loss reduction. Indeed, Wang et al. [50] show that performance degrades as the modified gradients diverge from the original gradients. The results in Figure 3a demonstrate that SDS-LoRA achieves superior loss convergence compared to preconditioning-based methods. Third, modifying the original gradients requires explicitly tracking optimizer momentum (e.g., in Adam). These methods introduce additional mechanisms to adjust the momentum based on the modified gradients, which may hinder practical adoption. In contrast, SDS-LoRA structurally decouples the influence of singular values from the gradients without modifying the original gradient, allowing straightforward adoption.

Optimality in the effective weight space does not determine the training trajectory. Theorem 3.1 characterizes the optimal effective gradient $\tilde{\boldsymbol{G}}$, that is, the update applied to $\boldsymbol{W}_{\text{eff}}$ at a single step given the current $\boldsymbol{A}$ and $\boldsymbol{B}$. Preconditioning-based methods attain the same optimum, and are therefore indistinguishable from SDS-LoRA by this criterion alone. This criterion, however, does not pin down the optimization trajectory. At the optimum, $\tilde{\boldsymbol{G}} = s^2(\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} + \boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})}\boldsymbol{G})$ is determined entirely by the two subspaces $\mathcal{R}(\boldsymbol{A})$ and $\mathcal{C}(\boldsymbol{B})$, and how much of $\boldsymbol{G}$ these subspaces capture is exactly the quantity α that governs the convergence rate in Theorem 3.2. These subspaces are in turn produced by the updates applied to $\boldsymbol{A}$ and $\boldsymbol{B}$, which differ across methods: preconditioning replaces the gradient of $\boldsymbol{A}$ with $\tilde{\nabla}_{\boldsymbol{A}}\mathcal{L} = s\boldsymbol{V}_{\boldsymbol{B}}\boldsymbol{\Sigma}_{\boldsymbol{B}}^{-1}\boldsymbol{U}_{\boldsymbol{B}}^\top\boldsymbol{G}$, which remains anisotropic and merely inverts the direction of the distortion, whereas SDS-LoRA yields the isometric $s\boldsymbol{Q}_{\boldsymbol{B}}^\top\boldsymbol{G}$. Two methods that agree on the one-step optimum in the effective weight space can thus drive $\mathcal{R}(\boldsymbol{A})$ and $\mathcal{C}(\boldsymbol{B})$ along very different paths. Figure 3b makes this concrete: although all the compared preconditioning-based methods attain the same optimal form of $\tilde{\boldsymbol{G}}$, SDS-LoRA sustains a markedly higher alignment between $\boldsymbol{G}$ and $\tilde{\boldsymbol{G}}$ throughout training, a gap that we attribute to the difference in the subspaces the methods

Table 13: **Mechanism-level comparison to gradient-based methods.** 'Invariant $\Delta \boldsymbol{W}$ update' indicates whether the update of $\Delta \boldsymbol{W}$ is independent of the scaling of $\boldsymbol{A}$ and $\boldsymbol{B}$, 'AGS in $\boldsymbol{A}, \boldsymbol{B}$ resolved' indicates whether anisotropic gradient scaling is removed from the updates of $\boldsymbol{A}$ and $\boldsymbol{B}$, and 'Original gradient' indicates whether the original gradients of $\boldsymbol{A}$ and $\boldsymbol{B}$ are used without modification. Among these methods, LoRA-Pro, ScaledAdamW, AltLoRA, and LoRA-RITE additionally attain the optimal form $\tilde{\boldsymbol{G}} = s^2(\boldsymbol{G}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} + \boldsymbol{P}_{\mathcal{C}(\boldsymbol{B})}\boldsymbol{G})$, and LoRA-GA attains it at the first training step only, whereas RefLoRA targets invariance to the choice of factorization rather than the gradient approximation quality itself.

Method	Scale-Invariant $\Delta \boldsymbol{W}$ update	AGS in $\boldsymbol{A}, \boldsymbol{B}$ resolved	Original gradient
LoRA-GA	✘	✘	✔
LoRA-Pro / ScaledAdamW / AltLoRA	✔	✘	✘
RefLoRA	✔	✘	✘
LoRA-RITE	✔	✘	✘
SDS-LoRA	✔	✔	✔

Table 14: **Comparison to LoRA-RITE and RefLoRA.** We report the average accuracy on the commonsense reasoning benchmarks with rank 32 from a single run. The reported numbers are taken from RefLoRA [60], whose experimental setup may differ from ours. In our reproduction, all aspects of the experimental setup follow ours except for the learning rate, which is searched over {1e-4, 2e-4, 5e-4, 1e-3}.

Method	Gemma-2B	LLaMA3-8B
LoRA-RITE (reported)	-	85.56
RefLoRA (reported)	-	85.75
LoRA-RITE (reproduced)	72.75	85.49
RefLoRA (reproduced)	73.97	85.83
SDS-LoRA	**74.80**	**86.08**

reach. The 'SDS-LoRA w/o detachment' variant in Table 15 isolates the same effect within our own formulation. In short, the two questions are distinct: Theorem 3.1 states which $\tilde{\boldsymbol{G}}$ one should aim for, whereas SDS-LoRA additionally governs the quality of the gradient that $\boldsymbol{A}$ and $\boldsymbol{B}$ actually receive on the way there.

Comparison to LoRA-RITE and RefLoRA. LoRA-RITE [54] and RefLoRA [60] address a closely related problem: the update of one low-rank factor is scaled by the magnitude of the other, which makes training sensitive to the particular factorization chosen.[4] RefLoRA observes that many pairs $(\boldsymbol{A}_t, \boldsymbol{B}_t)$ represent the same $\Delta \boldsymbol{W}$ and, at every step, selects the best balanced pair among them. It computes a matrix $\tilde{\boldsymbol{S}}_t$ from $(\boldsymbol{A}_t^\top \boldsymbol{A}_t)^{-1}$ and $\boldsymbol{B}_t^\top \boldsymbol{B}_t$, and preconditions the gradients fed to AdamW as $\boldsymbol{G}\boldsymbol{B}_t\tilde{\boldsymbol{S}}_t^{-1}$ for $\boldsymbol{A}$ and $\boldsymbol{G}^\top \boldsymbol{A}_t\tilde{\boldsymbol{S}}_t$ for $\boldsymbol{B}$, so that the resulting update to $\Delta \boldsymbol{W}$ is the same regardless of which valid pair one starts from. The gradient with respect to $\boldsymbol{A}$, however, still carries $\boldsymbol{B}_t$ and $\tilde{\boldsymbol{S}}_t$: in the balanced state $\boldsymbol{A}_t^\top \boldsymbol{A}_t = \boldsymbol{B}_t^\top \boldsymbol{B}_t$, which is exactly the state RefLoRA steers toward, $\tilde{\boldsymbol{S}}_t = \boldsymbol{I}$ and the preconditioned gradient reduces to LoRA's own $\boldsymbol{G}\boldsymbol{B}_t$, leaving anisotropic gradient scaling untouched. LoRA-RITE first builds a scale-free unmagnified gradient $\bar{\nabla}_{\boldsymbol{A}} = \nabla_{\boldsymbol{Z}}\boldsymbol{U}_{\boldsymbol{B}}$, where $\boldsymbol{U}_{\boldsymbol{B}}$ is obtained from the polar decomposition $\boldsymbol{B} = \boldsymbol{U}_{\boldsymbol{B}}\boldsymbol{R}_{\boldsymbol{B}}$, so that the magnitude of $\boldsymbol{B}$ is carried by $\boldsymbol{R}_{\boldsymbol{B}}$ and dropped. The applied update, however, is $\delta \boldsymbol{A}_t = -\eta \bar{\boldsymbol{M}}_{\boldsymbol{A}_t}\boldsymbol{R}_{\boldsymbol{B}}^{-\top}$, which reintroduces the magnitude of $\boldsymbol{B}$ through $\boldsymbol{R}_{\boldsymbol{B}}^{-\top}$. Since $\boldsymbol{R}_{\boldsymbol{B}}^{-\top}\boldsymbol{B}^\top = \boldsymbol{U}_{\boldsymbol{B}}^\top$, this reintroduced magnitude cancels once $\delta \boldsymbol{A}_t$ is folded into $\Delta \boldsymbol{W} = \delta \boldsymbol{A}_t \boldsymbol{B}^\top$, but the update applied to $\boldsymbol{A}$ itself still depends on the magnitude of $\boldsymbol{B}$, only inverted. Table 13 summarizes these mechanism-level differences, and Table 14 compares the fine-tuning performance. SDS-LoRA achieves the highest average accuracy on both models, supporting our claim that it is important to resolve anisotropic gradient scaling in the update of $\boldsymbol{A}$ and $\boldsymbol{B}$, not only in the update of $\Delta \boldsymbol{W}$.

[4] Both works write $\Delta \boldsymbol{W} = \boldsymbol{A}\boldsymbol{B}^\top$, so their $\boldsymbol{A}$ plays the role of our $\boldsymbol{B}$. We follow their own notation in this paragraph.

Table 15: **Comparison to orthogonalization-based LoRA variants.** 'Removes AGS' indicates whether anisotropic gradient scaling is removed throughout training, and 'Full RC of $\Delta \boldsymbol{W}$' indicates the full representational capacity of $\Delta \boldsymbol{W}$. We report the average accuracy on the commonsense reasoning benchmarks with Gemma-2B at rank 32 from a single run.

Method	Removes AGS	Full RC of $\Delta \boldsymbol{W}$	Avg. Accuracy
OLoRA	✘ (only at init.)	✔	71.24
QR-LoRA	✔	✘	4.67
QR-LoRA (trainable $\boldsymbol{Q}$)	✘	✔	69.88
SDS-LoRA w/o detachment	✘	✔	72.86
SDS-LoRA	✔	✔	**74.80**

Table 16: **Additional implementation details.**

	Full FT.		LoRA-based	
	Gemma-2B	LLaMA3-8B	Gemma-2B	LLaMA3-8B
Learning Rate	2e-5	1e-5	2e-4	1e-4
Learning Rate Scheduler	cosine scheduler			
Epochs	1			
Batch Size	32			
Target Modules	'q_proj', 'k_proj', 'v_proj', 'up_proj', 'down_proj', 'o_proj', 'gate_proj'			

I Comparison to Orthogonalization-Based LoRA Variants

Several LoRA variants also employ orthonormal bases obtained by QR decomposition. Our contribution, however, is not the use of orthonormal bases per se, but the structural decoupling of the singular values from the backward pass. We clarify this distinction at the mechanism level and validate it empirically.

OLoRA. OLoRA [8] applies QR decomposition once, at initialization, to the pretrained weight $\boldsymbol{W}_0 = \boldsymbol{Q}\boldsymbol{R}$, sets $\boldsymbol{B}$ to the first r columns of $\boldsymbol{Q}$ and $\boldsymbol{A}$ to the first r rows of $\boldsymbol{R}$, and then trains the standard LoRA formulation. Its backward pass is therefore identical to that of LoRA and still carries $\boldsymbol{\Sigma}_{\boldsymbol{B}}$. Since $\boldsymbol{B}$ is initialized with orthonormal columns, $\boldsymbol{\Sigma}\boldsymbol{B} = \boldsymbol{I}_r$ holds at the first iteration, but nothing constrains $\boldsymbol{\Sigma}\boldsymbol{B}$ once training proceeds. OLoRA thus addresses anisotropic gradient scaling only at initialization.

QR-LoRA. QR-LoRA [53] applies QR decomposition to the transpose of a rank-r core of the pretrained weight, $\boldsymbol{W}_{\text{core}}^{\top} = \boldsymbol{Q}\boldsymbol{R}$, and parameterizes the weight update as $\Delta \boldsymbol{W} = (\boldsymbol{Q}\Delta \boldsymbol{R})^{\top}$, where the orthonormal factor $\boldsymbol{Q}$ is frozen and only the increment $\Delta \boldsymbol{R}$ to the triangular factor is trained. Because the gradient reaches $\Delta \boldsymbol{R}$ only through the orthonormal $\boldsymbol{Q}$, anisotropic gradient scaling is indeed removed. However, the fixed $\boldsymbol{Q}$ confines the row space of $\Delta \boldsymbol{W}$ to a subspace determined by the pretrained weight, so QR-LoRA cannot represent an arbitrary rank-r update.

Comparison. As summarized in Table 15, neither method satisfies both desirable properties at once, whereas SDS-LoRA removes anisotropic gradient scaling from the backward pass without restricting the representational capacity of the forward pass. To further isolate the source of the gain, we evaluate two additional variants: QR-LoRA with a trainable $\boldsymbol{Q}$, which tests whether its fixed $\boldsymbol{Q}$ causes the collapse of QR-LoRA (4.67% in Table 15), and SDS-LoRA without detachment, which allows gradients to flow through the QR decomposition and thereby reintroduces anisotropic gradient scaling. SDS-LoRA and SDS-LoRA without detachment differ only in whether $\boldsymbol{Q}_{\boldsymbol{A}}$ and $\boldsymbol{Q}_{\boldsymbol{B}}$ are detached during the backward pass, and the gap between them indicates that the gain comes from removing anisotropic gradient scaling rather than from the use of orthonormal bases.

Broader Impacts

This research enhances the performance of LoRA, enabling low-rank adaptation to achieve results comparable to full fine-tuning. By improving the efficacy of parameter-efficient fine-tuning, our work allows high-quality model customization even in resource-constrained settings. This contributes to the development of more accurate and reliable AI systems across various domains. While improved performance could be utilized for unintended purposes, its primary impact lies in fostering robust and high-performing open-source models that do not require massive computational overhead.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Throughout the paper, we consistently claim that LoRA suffers from anisotropic gradient scaling (Figure 1 and Theorem 3.1) and our method addresses this successfully (Theorem 3.2).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in the "Limitations" paragraph following the conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We explicitly state the assumptions in Theorem 3.1 and Theorem 3.2, and provide the complete proof in Section A and Section B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide a detailed algorithm for the proposed method in Algorithm 1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is publicly available at `https://github.com/JungHunOh/SDS-LoRA_NeurIPS2026`, and all datasets used in our experiments are publicly available and cited in Section 4.1.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We provide experimental details in Section 4.1 and Table 16.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: For the main experimental results (Tables 1, 2, 3, 4, 5, 7, and 11), we report the mean and standard deviation across three independent runs. Some additional analyses in the appendix (Tables 14, 15, 8, and 9) are reported from a single run, as noted in their captions.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The specific GPU used for our experiments is described in Section 4.1. We provide an analysis of training overhead in Table 6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that this work strictly adheres to the Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impacts of our work in the "Broader Impacts" section at the end of the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate

to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the pre-trained models and datasets used in this study.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.