ORTHONORMAL REGULARIZATION IN LOW-RANK ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Performance degradation on tasks outside the fine-tuning domain is often observed while performing parameter-efficient fine-tuning (PEFT) on neural networks with limited data. For example, fine-tuning on mathematical datasets may impair the large language model's coding ability. We analyze this issue and identify the condition number of weight matrices as a key factor contributing to such degradation. To address this, we propose Singular Values and Orthonormal Regularized Singular Vectors Adaptation, or SORSA, a novel PEFT method that explicitly improves the conditioning of the adapted model parameters, thereby mitigating degradation and preserving broader capabilities. Empirically, we demonstrate that SORSA outperforms full fine-tuning, LoRA, PiSSA and AdaLoRA.

1 INTRODUCTION

Pre-trained large language models (LLMs) demonstrate strong generalization capabilities, enabling them to perform a wide range of natural language processing (NLP) tasks (Brown et al., 2020; Achiam et al., 2023; Touvron et al., 2023; Peng et al., 2024; Grattafiori et al., 2024). For adapting LLMs to specific downstream tasks, the default approach is often full parameter fine-tuning (Full FT), which updates all model parameters.

However, as LLMs continue to grow in scale, Full FT becomes increasingly impractical due to high computational and memory demands. To alleviate this, Parameter-Efficient Fine-Tuning (PEFT) methods have gained popularity, offering a cost-effective alternative by only updating a small subset of parameters.

Among PEFT approaches, LoRA (Hu et al., 2022) has emerged as a preferred choice due to its simplicity, efficiency, and minimal impact on inference-time latency. LoRA injects low-rank trainable matrices into the model, enabling effective fine-tuning with significantly reduced resource requirements.

Despite its efficiency, LoRA and similar PEFT methods face a major challenge under low-data regimes: they tend to overfit and degrade the model's original generalization ability, and even cause catastrophic forgetting (Xu et al., 2021a; Lin et al., 2024; Shuttleworth et al., 2024; van de Ven et al., 2024). For instance, fine-tuning on a small mathematical dataset may cause the model to forget previously acquired capabilities such as code generation or commonsense reasoning.

Previous works (Sinha et al., 2018; Saratchandran et al., 2024; Feng et al., 2025) have shown that neural networks with well-conditioned weight is able to provide a more robust performance. We further analyze this phenomenon in the context of PEFT, and identify the condition number of weight matrices as a critical factor affecting generalization during fine-tuning. Our study shows that LoRA often amplifies the condition number, making the adapted model increasingly ill-conditioned and unstable.

To address this, we propose a new PEFT method that explicitly improves the conditioning of the model during training. Our approach introduces orthonormal regularization to maintain well-conditioned weights, thereby preserving the model's generalization while enabling efficient adaptation. Empirical results show that our method significantly mitigates overfitting and outperforms existing baselines across various tasks.

We summarize our main contributions as follows:

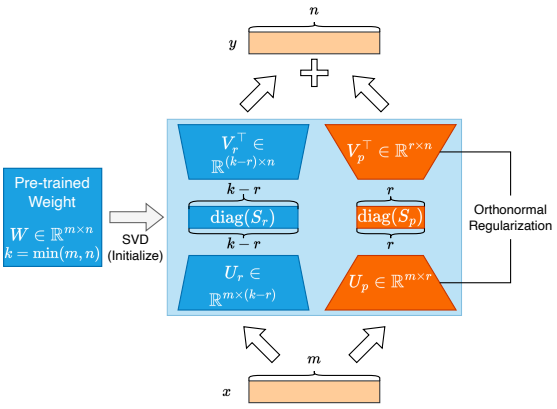

Figure 1: Illustration of SORSA.

- We demonstrate that during PEFT, well-conditioned weights tend to have better generalization.
- We propose SORSA, a novel parameter-efficient fine-tuning (PEFT) method that combines low-rank SVD-based initialization with orthonormal regularization.
- We provide the convergence rate of SORSA with gradient descent. (Theorem 5.4)
- We provide theoretical analysis showing that the orthonormal regularizer leads to better-conditioned weight updates. (Theorem 5.6)
- We empirically demonstrate that SORSA consistently outperforms or matches the performance of strong baselines, including full fine-tuning, LoRA, PiSSA, and AdaLoRA.

Roadmap. In Section 2, we present related work. In Section 3, we introduce the preliminary for our work. In Section 4, we propose our PEFT method. In Section 5, we provide theoretical analysis for SORSA. In Section 6, we conduct extensive experiments to validate SORSA's capability. In Section 7, we provide discussion and conclude the paper.

2 Related Work

Efficient Computation in Machine Learning. As the increasing scale of training data and model parameters, developing efficient machine learning algorithms have become central focus of recent AI research. In visual recognition, the acceleration of CNN (O'shea & Nash, 2015; He et al., 2016) and ViT (Dosovitskiy et al., 2020) have long been a heated topic, especially for edge devices that have limited computation resources. Representative acceleration techniques including architectural simplification (Sandler et al., 2018; Ding et al., 2021), quantization (Wu et al., 2016; Liu et al., 2021), and pruning (Yu et al., 2022). These techniques have significantly advance in real world applications, e.g. autonomous driving (Jiang et al., 2023b), medical image segmentation (Han et al., 2022), remote sensing (Xu et al., 2021b), emotion recognition (Zhang et al., 2021; Zhao et al., 2021; Liu et al., 2022a), and industrial automation. In content creation, diffusion models (Ho et al., 2020; Rombach et al., 2022) and flow matching models (Lipman et al., 2022; Liu et al., 2022c) are high-fidelity visual content generators. Acceleration in this area focuses on model architecture design (Dao et al., 2023; Frans et al., 2024; Chen et al., 2025; Cao et al., 2025a), fast ODE sampler (Xue et al., 2024b), complexity analysis (Gupta et al., 2024; Ke et al., 2025), distillation (Meng et al., 2023). These works have inspired many future applications, e.g. education, drug discovery (Wen et al., 2024), face synthesis (Liu et al., 2022b), and advertising (Liu et al., 2024a), and directions, e.g. benchmarks (Cao et al., 2025b; Guo et al., 2025a;b;c) and theoretical explorations (Hu et al., 2024). Graph Neural Networks are fundamental tools to model complex relational data (Veličković et al., 2018; Xu et al., 2019; Li et al., 2025), where important acceleration techniques include sparsification (Morris et al., 2020; Liu et al., 2023), GNN to MLP distillation (Zhang et al., 2022a; Han et al., 2023), and lazy computation (Narayanan et al., 2022; Zhang et al., 2024; Xue et al., 2024a). These techniques has inspired applications including but not limited to spatio-temporal data mining (Zhang et al., 2022b;

Wang et al., 2022), fake news detection (Xu et al., 2022; Chang et al., 2024), human skeleton-based visual recognition (Li et al., 2021; Fu et al., 2021), while also inspired aspects of graph neural networks including mitigating sensitive data influence (Chien et al., 2023; Zhang, 2024; Yi & Wei, 2025), and robustness (Geisler et al., 2021; Deng et al., 2022).

PEFT Methods. PEFT methods have been proposed to alleviate the inefficiency of full-parameter fine-tuning for large language models. These methods update only a small subset of parameters, often keeping the majority of the pre-trained model frozen, which significantly reduces memory and computational costs during training.

Adapter-based approaches were among the earliest PEFT methods, introduced by (Houlsby et al., 2019), where small trainable modules are inserted between frozen layers. Subsequent works such as (Lin et al., 2020) and (He et al., 2021) explored more compact or parallelized adapter designs. However, all adapter-based methods generally incur additional inference-time latency, since the inserted modules are not mergeable with the original model weights.

LoRA (Hu et al., 2022) gained popularity for introducing low-rank trainable matrices added to the pre-trained weight matrices. This approach avoids inference latency while offering competitive performance. Variants of LoRA expand upon this idea: AdaLoRA (Zhang et al., 2023) improves parameter efficiency by incorporating dynamic rank selection via singular value decomposition and pruning. DoRA (Liu et al., 2024b) decouples the direction and magnitude of weight updates, achieving higher expressiveness at the cost of higher training-time computation. OLoRA (Büyükakyüz, 2024) uses orthogonal initialization via QR decomposition to improve convergence speed. PiSSA (Meng et al., 2024) decomposes the pre-trained weight matrix and isolates a residual component, which remains frozen during training to improve convergence and stability.

Prompt-based PEFT methods, such as prefix-tuning (Lester et al., 2021), prepend learnable tokens to the model input. Although these methods are simple to implement, they often lead to longer input sequences and require careful prompt engineering. Other recent advances include GaLore (Zhao et al., 2024), which reduces memory usage through low-rank gradient accumulation, and LISA (Pan et al., 2024), which selectively fine-tunes critical layers using layer-wise importance sampling.

Condition Numbers in Neural Networks

3 PRELIMINARY

In this section, we first introduce our notations, then provide preliminary for our work.

3.1 NOTATIONS

We used $\mathbb{R}$ to denote set of real numbers. We use $A \in \mathbb{R}^{n \times d}$ to denote an $n \times d$ size matrix where each entry is a real number. We use I_d to denote the $d \times d$ identity matrix. We use $A^\top$ to denote the transpose of a matrix A. We use $A^{1/2}$ to denote element-wise square root of the matrix A, i.e. $(A^{1/2})_{i,j} = (A_{i,j})^{1/2}$. We use $\|A\|_F$ to denote Frobenius norm of matrix A. We use $\|A\|$ to denote spectral norm of matrix A. We use $A \preceq B$ to denote the positive semidefinite order, i.e. for symmetric $A, B \in \mathbb{R}^{d \times d}$, $A \preceq B \iff B - A \succeq 0$.

3.2 PEFT METHODS

LoRA LoRA (Hu et al., 2022) represents the weight as a low-rank decomposition:

$$W = W_0 + BA,$$

where $W_0 \in \mathbb{R}^{m \times n}$ is the frozen pre-trained weight, $A \in \mathbb{R}^{m \times r}$ is Gaussian-initialized, and $B \in \mathbb{R}^{r \times n}$ is initialized with zeros.

AdaLoRA. AdaLoRA (Zhang et al., 2023) introduces dynamic rank adaptation via SVD, and prunes less significant singular values to reduce parameter overhead.

DoRA. DoRA (Liu et al., 2024b) reformulates the weight update as a normalized decomposition:

$$W = m \cdot \frac{W_0 + BA}{\|W_0 + BA\|_c},$$

where $m = \|W_0 + BA\|_c$ is the column-wise norm. This improves model capacity but increases computational cost per step.

OLoRA. OLoRA (Büyükakyüz, 2024) initializes A and B using QR decomposition, ensuring orthonormality in the initial adapter weights, which empirically speeds up convergence.

PiSSA. PiSSA (Meng et al., 2024) decomposes W_0 via SVD as $W_0 = U\Sigma V^\top$ and splits it into:

$$W_{\text{pri}} = AB, \quad \text{where } A = U_p S_p^{1/2}, \quad B = S_p^{1/2} V_p^\top,$$

with U_p, S_p, V_p being the top-r components. The residual $W_{\text{res}} = U_r S_r V_r^\top$ remains frozen during training. This results in faster convergence and improved model fit.

3.3 Condition Number

We here provide a formal definition for the condition number.

Definition 3.1 (Condition Number)**.** *Let $A \in \mathbb{R}^{m\times n}$ be a matrix with full column rank. The* condition number *of A with respect to the spectral norm is defined as*

$$\kappa(A) := \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)} = \|A\| \cdot \|A^{-1}\|,$$

where $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ are the largest and smallest nonzero singular values of A.

4 Our Method

Giving a matrix $W \in \mathbb{R}^{m\times n}$, with $m \geq n$ (without loss of generality), we could perform SVD to decompose W by $W = U\,\mathrm{diag}(S)V^\top$. Here, $U \in \mathbb{R}^{m\times k}$ is a matrix of left singular vectors and has orthonormal columns, $V \in \mathbb{R}^{n\times k}$ is a matrix of right singular vectors and has orthonormal columns, and $S \in \mathbb{R}^k$ are singular values $\sigma^1, \sigma^2 \ldots \sigma^k$ arranged in descending order. $\mathrm{diag}(S)$ is constructed by placing the elements of $S \in \mathbb{R}^k$ along the main diagonal, with all other elements zero.

According to our SVD notations, given a rank r where $r \ll k$, we could perform the low-rank approximation by selecting the first r items on the diagonal of Σ, which is the first r most significant singular values, and also select the first r columns of U and first r rows of $V^\top$, which correspond to the selected singular values. By performing SVD low-rank approximation, we could get a low-rank matrix that preserves the largest significant values and vectors, containing the matrix's "most essential" data.

We use $\Sigma_p \in \mathbb{R}^{n\times n}$ to denote a diagonal matrix where first r entries are non-zero and all the remaining $n-r$ entries. Similarly, we use $\Sigma_r \in \mathbb{R}^{n\times n}$ to denote a diagonal matrix where first $n-r$ entries are non-zero and all the remaining r entries are zeros. Let $\Sigma = \Sigma_p + \Sigma_r$. Let SVD of W be $W = U\Sigma V^\top$.

Therefore, for a pre-trained weight $W_0 \in \mathbb{R}^{m\times n}$, we could split it based on its singular value into principal weight W_p and residual weight W_r,

$$W_p := \underbrace{U}_{m\times n}\underbrace{\Sigma_p}_{n\times n}\underbrace{V^\top}_{n\times n} \in \mathbb{R}^{m\times n}, \quad W_r := \underbrace{U}_{m\times n}\underbrace{\Sigma_r}_{n\times n}\underbrace{V^\top}_{n\times n} \in \mathbb{R}^{m\times n}.$$

Here, U represents the matrix of left singular vectors, S represents the singular values, $\mathrm{diag}(W)$ denotes a function to form a diagonal matrix from W, and V represents the matrix of right singular vectors. Since Σ_p is zeroed out in the last $n-r$ entries, and Σ_r is zeroed out in the first r entries, we can easily find low-rank equivalents of W_p and W_r. Specifically,

$$W_p = \underbrace{U_p}_{m\times r}\underbrace{S_p}_{r\times r}\underbrace{V_p^\top}_{r\times n}, \qquad W_r = \underbrace{U_r}_{m\times(n-r)}\underbrace{S_r}_{(n-r)\times(n-r)}\underbrace{V_r^\top}_{(n-r)\times n},$$

where U_p is the first r columns of U, S_p is the first r columns and rows of Σ_p, V_p is the first r columns of V, U_r is the last $n-r$ columns of U, S_r is the last $n-r$ columns and rows of Σ_r, V_r is the last $n-r$ column of V.

The initialization of W_r in SORSA is same as PiSSA (Meng et al., 2024). Nevertheless, unlike PiSSA which merge S_p with U_p and $V_p^\top$ into A and B by $A = U_p S_p^{1/2}$ and $B = S_p^{1/2} V_p^\top$, SORSA remains U_p, S_p, and $V_p^\top$ in separate weight. SORSA is defined by Eq. (1), initially equivalent to the pre-trained weight W_0. During training, W_r remains frozen, and only U_p, S_p, and $V_p^\top$ are updated.

SORSA is defined as:

$$\mathrm{SORSA}(x) := x(W_r + W_p) = xW_r + xU_p \,\mathrm{diag}(S_p) V_p^\top. \tag{1}$$

We adopt an orthonormal regularizer for U_p and V_p.

Definition 4.1 (Orthonormal regularizer)**.** *The orthonormal regularizer is defined as*

$$\mathcal{L}_{\mathrm{reg}}(U_p, V_p) := \|U_p^\top U_p - I_m\|_F^2 + \|V_p^\top V_p - I_n\|_F^2.$$

The regularizer could enhance their orthonormality during training. We discuss and verify its importance and effectiveness in 5.

Therefore, parameter updating of W_p in a SORSA adapter at training step t could be expressed as:

$$W_{p,t+1} = W_{p,t} - \eta_t \nabla_{W_{p,t}} \mathcal{L}_{\mathrm{train}} - \gamma_t \nabla_{W_{p,t}} \mathcal{L}_{\mathrm{reg}}. \tag{2}$$

At training step t, $\nabla_{W_{p,t}} \mathcal{L}_{\mathrm{train}}$ denotes the gradient of $\mathcal{L}_{\mathrm{train}}$ respect to $W_{p,t}$, and $\nabla_{W_{p,t}} \mathcal{L}_{\mathrm{reg}}$ denotes the gradient of the orthonormal regularizer loss $\mathcal{L}_{\mathrm{reg}}$ respect to $W_{p,t}$. η_t and γ_t are the learning rates for training loss and regularizer loss at step t, respectively.

We update the SORSA as the following for implementation simplicity

$$W_{p,t+1} = W_{p,t} - \eta_t \left(\nabla_{W_{p,t}} \mathcal{L}_{\mathrm{train}} + \frac{\gamma}{\eta_d} \nabla_{W_{p,t}} \mathcal{L}_{\mathrm{reg}} \right), \tag{3}$$

η_d is the maximum learning rate from the scheduler. This implementation allows us to use only one optimizer and scheduler to deal with two different learning rates separately.

5 Theoretical Analysis

5.1 Convergence Rate

We begin by analyzing the convergence behavior of gradient descent when applied to our objective function, which consists of a data-fitting loss L_{train} and our orthonormal regularizer $\mathcal{L}_{\mathrm{reg}}$.

Lemma 5.1 (Lipschitz continuity of $\mathcal{L}_{\mathrm{reg}}$)**.** *Suppose $\|U_p\|_F \le M_U$ and $\|V_p\|_F \le M_V$. Then $\mathcal{L}_{\mathrm{reg}}$ is Lipschitz continuous in the Frobenius norm:*

$$|\mathcal{L}_{\mathrm{reg}}(U_p^1, V_p^1) - \mathcal{L}_{\mathrm{reg}}(U_p^2, V_p^2)| \le L_{\mathrm{reg}}(\|U_p^1 - U_p^2\|_F + \|V_p^1 - V_p^2\|_F),$$

where

$$L_{\mathrm{reg}} = 4M_U(M_U^2 + 1) + 4M_V(M_V^2 + 1).$$

Proof. Compute the partial gradients

$$\nabla_{U_p} \mathcal{L}_{\mathrm{reg}} = 4U_p(U_p^\top U_p - I_m),$$
$$\nabla_{V_p} \mathcal{L}_{\mathrm{reg}} = 4V_p(V_p^\top V_p - I_n).$$

Hence

$$\begin{aligned}\|\nabla \mathcal{L}_{\mathrm{reg}}\|_F &\le 4\|U_p\|\|U_p^\top U_p - I_m\|_F + 4\|V_p\|\|V_p^\top V_p - I_n\|_F \\ &\le 4M_U(M_U^2 + 1) + 4M_V(M_V^2 + 1).\end{aligned}$$

By the mean value theorem for vector functions,

$$|\mathcal{L}_{\mathrm{reg}}(X) - \mathcal{L}_{\mathrm{reg}}(Y)| \le \max_Z \|\nabla \mathcal{L}_{\mathrm{reg}}(Z)\|_F \|X - Y\|_F,$$

and the claimed bound follows. □

We now make two standard assumptions to ensure well-behaved optimization.

Assumption 5.2 (Smoothness and strong convexity of L_{train})**.** *The data term $L_{\text{train}}(W_p)$ is twice differentiable, μ_{train}-strongly convex and L_{train}-smooth:*

$$\mu_{\text{train}} I \preceq \nabla^2 L_{\text{train}}(W) \preceq L_{\text{train}} I \quad \textit{for all } W_p.$$

Assumption 5.3 (Hessian lower bound for $\mathcal{L}_{\text{reg}}$)**.** *There is a constant $C_{\text{reg}} \geq 0$ such that*

$$\nabla^2 \mathcal{L}_{\text{reg}}(W) \succeq -C_{\text{reg}} I \quad \textit{for all } W = (U_p, V_p).$$

The next theorem establishes that, under these assumptions, SORSA converges linearly.

Theorem 5.4 (Linear convergence of SORSA)**.** *Let*

$$F(W_p) = L_{\text{train}}(W_p) + \gamma \mathcal{L}_{\text{reg}}(W_p),$$

and suppose Assumptions 5.2 and 5.3 hold. If

$$0 < \gamma < \frac{\mu_{\text{train}}}{C_{\text{reg}}}, \quad \eta \in (0, \frac{2}{L_{\text{train}} + \gamma L_{\text{reg}}}),$$

then gradient descent

$$W_p^{t+1} = W_p^t - \eta \nabla F(W_p^t)$$

satisfies

$$F(W_p^t) - F(W_p^*) \leq (1 - \eta(\mu_{\text{train}} - \gamma C_{\text{reg}}))^t (F(W_p^0) - F(W_p^*)).$$

In particular, setting $\eta = 1/(L_{\text{train}} + \gamma L_{\text{reg}})$ gives

$$F(W_p^t) - F(W_p^*) \leq (1 - \frac{\mu_{\text{train}} - \gamma C_{\text{reg}}}{L_{\text{train}} + \gamma L_{\text{reg}}})^t (F(W_p^0) - F(W_p^*)).$$

Proof. By Assumption 5.2, $\nabla^2 L_{\text{train}} \geq \mu_{\text{train}} I$ and by Assumption 5.3, $\nabla^2(\gamma \mathcal{L}_{\text{reg}}) \geq -\gamma C_{\text{reg}} I$. Hence,

$$\nabla^2 F = \nabla^2 L_{\text{train}} + \gamma \nabla^2 \mathcal{L}_{\text{reg}} \geq (\mu_{\text{train}} - \gamma C_{\text{reg}}) I,$$

and also $\nabla^2 F \leq (L_{\text{train}} + \gamma L_{\text{reg}}) I$. The claimed rate follows from standard gradient descent guarantees. □

5.2 Condition Number

We now analyze how the regularizer in SORSA helps maintain a smaller condition number for the weight matrix. A well-conditioned weight matrix is essential for stable optimization and good generalization.

We begin with a lemma that shows the singular values of the regularized weight matrix stay close to those of the unregularized one, provided the regularizer gradient is small.

Lemma 5.5. *Let*

$$W_p^{\text{unreg},t} = U_p^{\text{unreg},t} S_p^{\text{unreg},t} (V_p^{\text{unreg},t})^\top, \quad W_p^{\text{reg},t} = U_p^{\text{reg},t} S_p^{\text{reg},t} (V_p^{\text{reg},t})^\top$$

be the outputs of one step of SORSA at step t with and without regularizer, respectively.

If $\|\nabla_{W_p} \mathcal{L}_{\text{reg}}\|_F \leq \epsilon_\nabla$, then for each singular value σ_i,

$$(1 - \epsilon)\sigma_i^{\text{unreg},t} \leq \sigma_i^{\text{reg},t} \leq (1 + \epsilon)\sigma_i^{\text{unreg},t},$$

where $\epsilon = \gamma \epsilon_\nabla$.

Proof. We have

$$W_p^{\text{reg}} - W_p^{\text{unreg},t} = \gamma \nabla_{W_p} \mathcal{L}_{\text{reg}}, \quad \|W_p^{\text{reg}} - W_p^{\text{unreg},t}\|_F = \gamma \epsilon_\nabla.$$

By Weyl's inequality,

$$|\sigma_i^{\text{reg},t} - \sigma_i^{\text{unreg},t}| \leq \|W_p^{\text{reg},t} - W_p^{\text{unreg},t}\| \leq \|W_p^{\text{reg},t} - W_p^{\text{unreg},t}\|_F \leq \gamma \epsilon_\nabla.$$

The last inequality follows directly. □

We now prove our main theorem: the condition number of the regularized weight matrix is strictly smaller than that of the unregularized one.

Theorem 5.6. *Under the setup of Lemma 5.5, assume that $\nabla\mathcal{L}_{\text{train}}$ is invariant for all $t > 0$. Let the orthonormal regularizer be defined in Definition 4.1 Then for every iteration $t > 0$,*

$$\kappa(W_p^{\text{reg},t}) < \kappa(W_p^{\text{unreg},t}),$$

where κ is defined in Definition 3.1.

Proof. We divide the proof into four steps to illustrate how regularization improves conditioning.

Step 1. Factor-wise bounds. For any factorization $W = USV^\top$ with diagonal S,

$$\|W\| \le \|U\|\|S\|\|V\|, \quad \|W^{-1}\| \le \|V\|\|S^{-1}\|\|U^{-1}\|.$$

Hence,

$$\kappa(W) \le \kappa(U)\kappa(S)\kappa(V).$$

Step 2. Singular value perturbation. According to Lemma 5.5,

$$|\sigma_i^{\text{reg},t} - \sigma_i^{\text{unreg},t}| \le \epsilon_t,$$

which implies

$$\kappa(S_p^{\text{reg},t}) \le \frac{1+\epsilon_t}{1-\epsilon_t}\kappa(S_p^{\text{unreg},t}).$$

Step 3. Orthonormal regularizer bounds factor condition numbers. By definition of $\mathcal{L}_{\text{reg}}$ in Definition 4.1, and $\nabla\mathcal{L}_{\text{train}}$ is invariant for all $t > 0$,

$$\kappa(U_p^{\text{reg},t}) < \kappa(U_p^{\text{unreg},t}), \quad \kappa(V_p^{\text{reg},t}) < \kappa(V_p^{\text{unreg},t}).$$

Step 4: Combine bounds to compare condition numbers. By the above,

$$\begin{aligned}\kappa(W_p^{\text{reg},t}) &\le \kappa(U_p^{\text{reg},t})\kappa(S_p^{\text{reg},t})\kappa(V_p^{\text{reg},t}) \\ &\le \kappa(U_p^{\text{reg},t})\kappa(V_p^{\text{reg},t})\frac{1+\epsilon_t}{1-\epsilon_t}\kappa(S_p^{\text{unreg},t}),\end{aligned}$$

and

$$\kappa(W_p^{\text{unreg},t}) \ge \kappa(U_p^{\text{unreg},t})\kappa(S_p^{\text{unreg},t})\kappa(V_p^{\text{unreg},t}).$$

So,

$$\frac{\kappa(W_p^{\text{reg},t})}{\kappa(W_p^{\text{unreg},t})} \le \frac{\kappa(U_p^{\text{reg},t})\kappa(V_p^{\text{reg},t})}{\kappa(U_p^{\text{unreg},t})\kappa(V_p^{\text{unreg},t})} \cdot \frac{1+\epsilon_t}{1-\epsilon_t} < 1.$$

Thus, $\kappa(W_p^{\text{reg},t}) < \kappa(W_p^{\text{unreg},t})$, completing the proof. □

6 Experiments

We conducted comparative experiments on different NLP tasks, including natural language generation (NLG) between SORSA, PiSSA (Meng et al., 2024), LoRA (Hu et al., 2022), AdaLoRA (Zhang et al., 2023), and full parameter fine-tuning.

We conducted NLG tests on Llama 2 7B (Touvron et al., 2023), RWKV6 7B (Peng et al., 2024), Mistral 7B v0.1 (Jiang et al., 2023a) and Gemma 7B (Gemma Team, 2024). We trained the models using the first 100K data in MetaMathQA (Yu et al., 2023) and evaluated the model on GSM-8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). We also trained the model on the first 100K data in CodeFeedback Filtered Instruction (Zheng et al., 2024) dataset and evaluated it on HumanEval (Chen et al., 2021). The training process followed identical setups as the experiments conducted in PiSSA (Meng et al., 2024). All reported values are accuracy in percentage. See

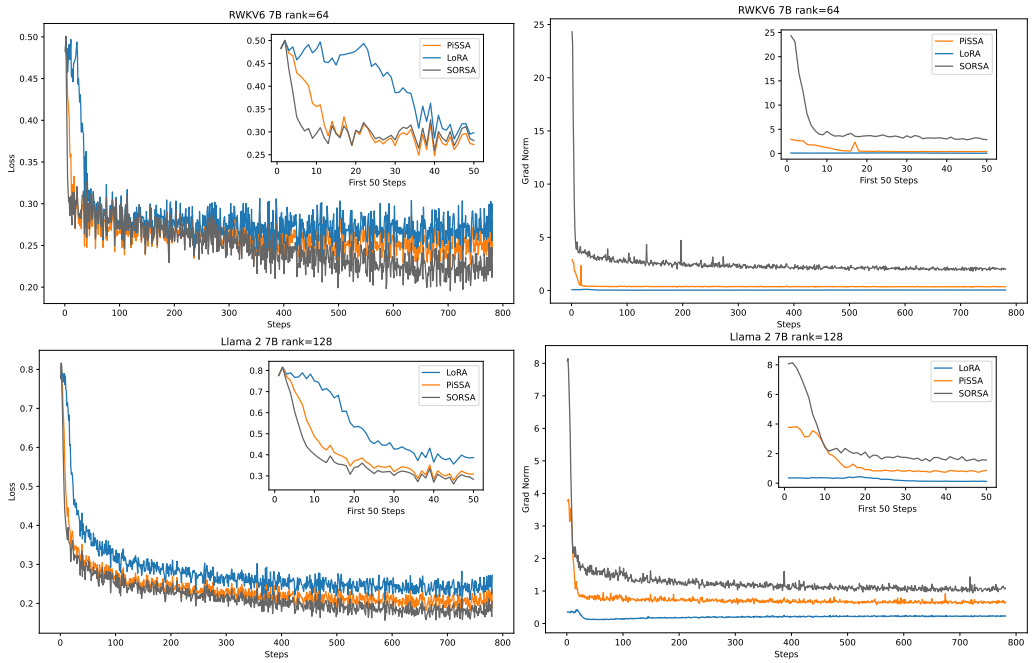

Figure 2: The training loss and gradient norm comparison between SORSA, PiSSA, and LoRA on MetaMathQA training of RWKV6 7B and Llama 2 7B. LoRA and PiSSA curves of Llama 2 7B are from (Meng et al., 2024).

Table 1: Comparing SORSA with other methods on NLG tasks. † denotes results from (Meng et al., 2024). We use **TPara.** to represent trainable parameters.

Model	Method	TPara.	GSM-8K	MATH	HumanEval
Llama 2 7B	Full FT	6738M	49.05†	7.22†	21.34†
Llama 2 7B	LoRA	320M	42.30†	5.50†	18.29†
Llama 2 7B	PiSSA	320M	53.07†	7.44†	21.95†
Llama 2 7B	AdaLoRA	320M	47.30	6.48	19.51
Llama 2 7B	SORSA	320M	**56.03**	**10.36**	**24.39**
RWKV6 7B	LoRA	176M	8.04†	7.38	15.24
RWKV6 7B	PiSSA	176M	32.07	9.42	17.07
RWKV6 7B	AdaLoRA	176M	33.28	8.08	15.85
RWKV6 7B	SORSA	176M	**45.87**	**11.32**	**22.56**
Mistral 7B	Full FT	7242M	67.02†	18.60†	45.12†
Mistral 7B	LoRA	168M	67.70†	19.68†	43.90†
Mistral 7B	PiSSA	168M	72.86†	21.54†	46.95†
Mistral 7B	AdaLoRA	168M	72.25	21.06	45.73
Mistral 7B	SORSA	168M	**73.09**	**21.86**	**47.56**
Gemma 7B	Full FT	8538M	71.34†	22.74†	46.95†
Gemma 7B	LoRA	200M	74.90†	31.28†	53.66†
Gemma 7B	PiSSA	200M	77.94†	**31.94** †	54.27†
Gemma 7B	AdaLoRA	200M	**78.99**	31.44	**55.49**
Gemma 7B	SORSA	200M	78.09	29.52	**55.49**

Section A for more details and hyperparameters of the training. We quoted some PiSSA, LoRA, and full parameter fine-tuning results from (Meng et al., 2024). Some of our experiments were conducted on a single NVIDIA A100-SXM4 (80GB) GPU, and others were conducted on a single NVIDIA H100-SXM4 (80GB) GPU. See Table 1 for the results and Figure 2 for the loss and gradient norm comparison.

The results showed that across all models tested, SORSA generally outperformed other methods, though with some notable exceptions. For mathematical evaluations on Llama 2 7B, SORSA scored 56.03% on GSM-8K and 10.36% on MATH, significantly outperforming other methods. For the RWKV6 7B model, SORSA achieved 45.87% accuracy on GSM-8K and 11.32% on MATH, surpassing both PiSSA and AdaLoRA, with AdaLoRA showing competitive performance on GSM-8K at 33.28%. On Mistral 7B, SORSA reached 73.09% on GSM-8K and 21.86% on MATH, showing modest improvements over AdaLoRA's strong performance of 72.25% and 21.06%, respectively. With Gemma 7B, the results were mixed - while AdaLoRA achieved the highest GSM-8K score at 78.99% and competitive MATH performance at 31.44%, SORSA maintained strong performance with 78.09% on GSM-8K. However, its MATH score of 29.52% was lower than other methods. In coding evaluations, SORSA and AdaLoRA showed strong performance on HumanEval, with both methods achieving 55.49% on Gemma 7B, while SORSA maintained an edge across other model variants. Additionally, we did not include loss and gradient norm curves in our figure because the regularizer in AdaLoRA and Gaussian initialization caused significantly higher initial loss values, making direct comparisons with other methods inappropriate.

The Figure 2 reveals that SORSA and PiSSA exhibit nearly identical loss curves at the beginning and even slightly higher than PiSSA on RWKV-6 training. However, when the training step is approximately $t > 300$, SORSA steadily decreases its loss. In contrast, LoRA and PiSSA show a deceleration in their loss reduction. The observations on loss curves are also valid for the changing rate of gradient norm, where SORSA showed a more consistent decrease in gradient norm compared to LoRA and PiSSA. This can be explained by Theorem 5.6, especially at later stages of training.

7 Discussion and Conclusion

In this paper, we introduced SORSA, a novel parameter-efficient fine-tuning (PEFT) method designed to enhance the adaptation of large language models (LLMs) for downstream tasks. SORSA utilizes singular value decomposition (SVD) to split pre-trained weights into principal and residual components, only training the principal singular values and vectors while freezing the residuals. We implemented an orthonormal regularizer to maintain the orthonormality of singular vectors during training, ensuring efficient parameter updates and preserving the integrity of singular values.

Our experiments demonstrated that SORSA outperforms existing PEFT methods, such as LoRA and PiSSA, in both convergence speed and accuracy on the NLG tasks. Specifically, Llama 2 7B, tuned with SORSA, achieved significant improvements in the GSM-8K and MATH benchmarks, highlighting the effectiveness of our approach.

We adopted singular values and vector analysis, comparing SORSA with FT and LoRA. SORSA is superior in preserving the pre-trained weight's singular values and vectors during training. This suggests an explanation for SORSA's supreme performance demonstrated in the experiment. We also show the significance of the orthonormal regularizer through analysis.

Our theoretical analysis provided a mathematical foundation for SORSA, demonstrating its convexity, Lipschitz continuity, and the crucial role of the regularizer in improving the optimization landscape. This theoretical framework explains SORSA's empirical superior performance and offers valuable insights for future developments in adaptive learning algorithms.

SORSA retains the advantages of LoRA and variants, including low training VRAM requirements, no inference latency, and versatility across different neural network architectures. By offering a more efficient fine-tuning mechanism, SORSA presents a promising direction for future research and application in the field of LLMs.

Overall, SORSA gives a new perspective on parameter-efficient fine-tuning, showcasing exceptional efficiency and robust performance. It outperforms existing methods like LoRA and PiSSA in several downstream tasks and maintains the practical benefits of low VRAM requirements, no inference latency, and ease of implementation. This innovative approach offers a promising direction of singular values and vector analysis for future research and practical applications in adapting pre-trained models, making it a pivotal development in the field.

[1]This significant under-perform due to LoRA failed to learn the GSM-8K required answer formatting behavior.

Ethic Statement

This paper does not involve human subjects, personally identifiable data, or sensitive applications. We do not foresee direct ethical risks. We follow the ICLR Code of Ethics and affirm that all aspects of this research comply with the principles of fairness, transparency, and integrity.

Reproducibility Statement

We ensure reproducibility on both theoretical and empirical fronts. For theory, we include all formal assumptions, definitions, and complete proofs in the appendix. For experiments, we describe model architectures, datasets, preprocessing steps, hyperparameters, and training details in the main text and appendix. Code and scripts are provided in the supplementary materials to replicate the empirical results.

References

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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

Yang Cao, Bo Chen, Xiaoyu Li, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Mingda Wan. Force matching with relativistic constraints: A physics-inspired approach to stable and efficient generative modeling. *arXiv preprint arXiv:2502.08150*, 2025a.

Yuefan Cao, Xuyang Guo, Jiayan Huo, Yingyu Liang, Zhenmei Shi, Zhao Song, Jiahao Zhang, and Zhen Zhuang. Text-to-image diffusion models cannot count, and prompt refinement cannot help. *arXiv preprint arXiv:2503.06884*, 2025b.

Ya-Ting Chang, Zhibo Hu, Xiaoyu Li, Shuiqiao Yang, Jiaojiao Jiang, and Nan Sun. Dihan: A novel dynamic hierarchical graph attention network for fake news detection. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 197–206, 2024.

Bo Chen, Chengyue Gong, Xiaoyu Li, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Mingda Wan. High-order matching for one-step shortcut diffusion models. *arXiv preprint arXiv:2502.00688*, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Eli Chien, Chao Pan, and Olgica Milenkovic. Efficient model updates for approximate unlearning of graph-structured data. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=fhcu4FBLciL.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

Chenhui Deng, Xiuyu Li, Zhuo Feng, and Zhiru Zhang. GARNET: Reduced-rank topology learning for robust and scalable graph neural networks. In *The First Learning on Graphs Conference*, 2022. URL https://openreview.net/forum?id=kvwWjYQtmw.

Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13733–13742, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yangqi Feng, Shing-Ho J Lin, Baoyuan Gao, and Xian Wei. Lipschitz constant meets condition number: Learning robust and compact deep neural networks. *arXiv preprint arXiv:2503.20454*, 2025.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.

Ziwang Fu, Feng Liu, Jiahao Zhang, Hanyang Wang, Chengyi Yang, Qing Xu, Jiayin Qi, Xiangling Fu, and Aimin Zhou. Sagn: semantic adaptive graph network for skeleton-based human action recognition. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pp. 110–117, 2021.

Simon Geisler, Tobias Schmidt, Hakan Sirin, Daniel Zügner, Aleksandar Bojchevski, and Stephan Günnemann. Robustness of graph neural networks at scale. In *NeurIPS*, 2021.

Google DeepMind Gemma Team. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Xuyang Guo, Zekai Huang, Jiayan Huo, Yingyu Liang, Zhenmei Shi, Zhao Song, and Jiahao Zhang. Can you count to nine? a human evaluation benchmark for counting limits in modern text-to-video models. *arXiv preprint arXiv:2504.04051*, 2025a.

Xuyang Guo, Jiayan Huo, Zhenmei Shi, Zhao Song, Jiahao Zhang, and Jiale Zhao. T2vphysbench: A first-principles benchmark for physical consistency in text-to-video generation. *arXiv preprint arXiv:2505.00337*, 2025b.

Xuyang Guo, Jiayan Huo, Zhenmei Shi, Zhao Song, Jiahao Zhang, and Jiale Zhao. T2vtextbench: A human evaluation benchmark for textual control in video generation models. *arXiv preprint arXiv:2505.04946*, 2025c.

Shivam Gupta, Aditya Parulekar, Eric Price, and Zhiyang Xun. Improved sample complexity bounds for diffusion model training. *Advances in Neural Information Processing Systems*, 37:40976–41012, 2024.

Xiaotian Han, Tong Zhao, Yozen Liu, Xia Hu, and Neil Shah. MLPInit: Embarrassingly simple GNN training acceleration with MLP initialization. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=P8YIphWNEGO.

Zhimeng Han, Muwei Jian, and Gai-Ge Wang. Convunext: An efficient convolution neural network for medical image segmentation. *Knowledge-based systems*, 253:109512, 2022.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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. *arXiv preprint arXiv:2103.03874*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Jerry Yao-Chieh Hu, Weimin Wu, Zhuoru Li, Sophia Pi, Zhao Song, and Han Liu. On statistical rates and provably efficient criteria of latent diffusion transformers (dits). *Advances in Neural Information Processing Systems*, 37:31562–31628, 2024.

Albert Q Jiang, A Sablayrolles, A Mensch, C Bamford, DS Chaplot, D de las Casas, F Bressand, G Lengyel, G Lample, L Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023a.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8350, 2023b.

Yekun Ke, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Circuit complexity bounds for visual autoregressive model. *arXiv preprint arXiv:2501.04299*, 2025.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction. *IEEE transactions on pattern analysis and machine intelligence*, 44(6):3316–3333, 2021.

Xiaoyu Li, Yingyu Liang, Zhenmei Shi, Zhao Song, Wei Wang, and Jiahao Zhang. On the computational capability of graph neural networks: A circuit complexity bound perspective. *arXiv preprint arXiv:2501.06444*, 2025.

Yang Lin, Xinyu Ma, Xu Chu, Yujie Jin, Zhibang Yang, Yasha Wang, and Hong Mei. Lora dropout as a sparsity regularizer for overfitting control. *arXiv preprint arXiv:2404.09610*, 2024.

Zhaojiang Lin, Andrea Madotto, and Pascale Fung. Exploring versatile generative language model via parameter-efficient transfer learning. *arXiv preprint arXiv:2004.03829*, 2020.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Chengyi Liu, Jiahao Zhang, Shijie Wang, Wenqi Fan, and Qing Li. Score-based generative diffusion models for social recommendations. *arXiv preprint arXiv:2412.15579*, 2024a.

Feng Liu, Han-Yang Wang, Si-Yuan Shen, Xun Jia, Jing-Yi Hu, Jia-Hao Zhang, Xi-Yi Wang, Ying Lei, Ai-Min Zhou, Jia-Yin Qi, et al. Opo-fcm: a computational affection based occ-pad-ocean federation cognitive modeling approach. *IEEE Transactions on Computational Social Systems*, 10(4):1813–1825, 2022a.

Feng Liu, Hanyang Wang, Jiahao Zhang, Ziwang Fu, Aimin Zhou, Jiayin Qi, and Zhibin Li. Evogan: An evolutionary computation assisted gan. *Neurocomputing*, 469:81–90, 2022b.

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 *Forty-first International Conference on Machine Learning*, 2024b.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022c.

Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34:28092–28103, 2021.

Zirui Liu, Shengyuan Chen, Kaixiong Zhou, Daochen Zha, Xiao Huang, and Xia Hu. Rsc: Accelerate graph neural networks training via randomized sparse computations. *ICML*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14297–14306, 2023.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37:121038–121072, 2024.

Christopher Morris, Gaurav Rattan, and Petra Mutzel. Weisfeiler and leman go sparse: Towards scalable higher-order graph embeddings. In *NeurIPS*, 2020.

S Deepak Narayanan, Aditya Sinha, Prateek Jain, Purushottam Kar, and SUNDARARAJAN SELLAMANICKAM. IGLU: Efficient GCN training via lazy updates. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=5kq11Tl1z4.

Keiron O'shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. Lisa: layerwise importance sampling for memory-efficient large language model fine-tuning. *Advances in Neural Information Processing Systems*, 37:57018–57049, 2024.

Bo Peng, Daniel Goldstein, Quentin Anthony, Alon Albalak, Eric Alcaide, Stella Biderman, Eugene Cheah, Teddy Ferdinan, Haowen Hou, Przemysław Kazienko, et al. Eagle and finch: Rwkv with matrix-valued states and dynamic recurrence. *arXiv preprint arXiv:2404.05892*, 3, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Hemanth Saratchandran, Thomas X Wang, and Simon Lucey. Weight conditioning for smooth optimization of neural networks. In *European Conference on Computer Vision*, pp. 310–325. Springer, 2024.

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. Lora vs full fine-tuning: An illusion of equivalence. *arXiv preprint arXiv:2410.21228*, 2024.

Abhishek Sinha, Mayank Singh, and Balaji Krishnamurthy. Neural networks in an adversarial setting and ill-conditioned weight space. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 177–190. Springer, 2018.

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

Gido M van de Ven, Nicholas Soures, and Dhireesha Kudithipudi. Continual learning and catastrophic forgetting. *arXiv preprint arXiv:2403.05175*, 2024.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Lijing Wang, Aniruddha Adiga, Jiangzhuo Chen, Adam Sadilek, Srinivasan Venkatramanan, and Madhav Marathe. Causalgnn: Causal-based graph neural networks for spatio-temporal epidemic forecasting. In *AAAI*, 2022.

Yibo Wen, Chenwei Xu, Jerry Yao-Chieh Hu, and Han Liu. Alignab: Pareto-optimal energy alignment for designing nature-like antibodies. *arXiv preprint arXiv:2412.20984*, 2024.

Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4820–4828, 2016.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. Raise a child in large language model: Towards effective and generalizable fine-tuning. *arXiv preprint arXiv:2109.05687*, 2021a.

Weizhi Xu, Junfei Wu, Qiang Liu, Shu Wu, and Liang Wang. Evidence-aware fake news detection with graph neural networks. In *Proceedings of the ACM web conference 2022*, pp. 2501–2510, 2022.

Zhiyong Xu, Weicun Zhang, Tianxiang Zhang, Zhifang Yang, and Jiangyun Li. Efficient transformer for remote sensing image segmentation. *Remote Sensing*, 13(18):3585, 2021b.

Rui Xue, Haoyu Han, Mohamadali Torkamani, Jian Pei, and Xiaorui Liu. Lazygnn: Large-scale graph neural networks via lazy propagation. In *ICML*, 2024a.

Shuchen Xue, Zhaoqiang Liu, Fei Chen, Shifeng Zhang, Tianyang Hu, Enze Xie, and Zhenguo Li. Accelerating diffusion sampling with optimized time steps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8292–8301, 2024b.

Lu Yi and Zhewei Wei. Scalable and certifiable graph unlearning: Overcoming the approximation error barrier. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pPyJyeLriR.

Fang Yu, Kun Huang, Meng Wang, Yuan Cheng, Wei Chu, and Li Cui. Width & depth pruning for vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3143–3151, 2022.

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. *arXiv preprint arXiv:2309.12284*, 2023.

Jiahao Zhang. Graph unlearning with efficient partial retraining. In *Companion Proceedings of the ACM on Web Conference 2024*, pp. 1218–1221, 2024.

Jiahao Zhang, Feng Liu, and Aimin Zhou. Off-tanet: A lightweight neural micro-expression recognizer with optical flow features and integrated attention mechanism. In *Pacific Rim International Conference on Artificial Intelligence*, pp. 266–279. Springer, 2021.

Jiahao Zhang, Rui Xue, Wenqi Fan, Xin Xu, Qing Li, Jian Pei, and Xiaorui Liu. Linear-time graph neural networks for scalable recommendations. In *Proceedings of the ACM on Web Conference 2024*, pp. 3533–3544, 2024.

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. *arXiv preprint arXiv:2303.10512*, 2023.

Shichang Zhang, Yozen Liu, Yizhou Sun, and Neil Shah. Graph-less neural networks: Teaching old MLPs new tricks via distillation. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=4p6_5HBWPCw.

Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Zhou Qin, and Wenwu Zhu. Dynamic graph neural networks under spatio-temporal distribution shift. In *NeurIPS*, 2022b.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024.

Zengqun Zhao, Qingshan Liu, and Feng Zhou. Robust lightweight facial expression recognition network with label distribution training. In *AAAI*, 2021.

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.

Appendix

Roadmap. In the appendix, we present the experiments details in Section A.

A Experiments Details

For our NLG tasks, we adapted Llama 2 7B (Touvron et al., 2023), RWKV6 7B (Peng et al., 2024), Mistral 7B v0.1 (Jiang et al., 2023a) Gemma 7B (Gemma Team, 2024) models by SORSA. For GSM-8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) evaluations, we trained those models with the first 100K data in MetaMathQA (Yu et al., 2023) dataset. For HumanEval (Chen et al., 2021) evaluation, we use the first 100K data in CodeFeedback Filtered Instruction (Zheng et al., 2024) dataset.

We used AdamW (Loshchilov & Hutter, 2017) optimizer and cosine annealing scheduler in training. SORSA adapters were applied on all linear matrices in every layer. We only calculated the loss on the response part. The models are loaded in FP32 and trained with TF32 & BF16 mix precision. In our experiments, we selected a higher learning rate for SORSA than other methods to counterbalance the negative effect of orthonormal regularizer on optimizing toward lower training loss. See Table 2 and 3 for hyperparameters.

Table 2: Hyperparameters for training with SORSA, LoRA and PiSSA on different models for GSM-8K and MATH

Model	Llama 2 7B	RWKV6 7B	RWKV6 7B	Mistral 7B	Gemma 7B
Method	SORSA	SORSA	LoRA&PiSSA	SORSA	SORSA
Mix-Precision	TF32+BF16	TF32+BF16	TF32+BF16	TF32+BF16	TF32+BF16
Epoch	1	1	1	1	1
Batch Size	128	128	128	128	128
Max Length	512	512	512	512	512
Weight Decay	0	0	0	0	0
Warm-up Ratio	0.03	0.03	0.03	0.03	0.03
Learning Rate	3e-5	3e-5	2e-5	3e-5	3e-5
Grad Clip	1.0	1.0	1.0	1.0	1.0
SORSA γ	4e-4	4e-4	N/A	4e-4	4e-4
Rank	128	64	64	64	64

Table 3: Hyperparameters for evaluation with SORSA, LoRA and PiSSA on different models for GSM-8K and MATH. ML denotes Max Length.

Model	Llama 2 7B	RWKV6 7B	RWKV6 7B	Mistral 7B	Gemma 7B
Method	SORSA	SORSA	LoRA & PiSSA	SORSA	SORSA
Precision	BF16	FP32	FP32	BF16	BF16
Sampling	False	False	False	False	False
Top-P	1.0	1.0	1.0	1.0	1.0
ML for GSM-8K	1024	1024	1024	1024	1024
ML for MATH	2048	2048	2048	2048	2048
ML for HumanEval	2048	2048	2048	2048	2048

LLM Usage Disclosure

LLMs were used only to polish language, such as grammar and wording. These models did not contribute to idea creation or writing, and the authors take full responsibility for this paper's content.

Table 4: Hyperparameters of training for with AdaLoRA on different models for GSM-8K and MATH

Model	Llama 2 7B	Mistral 7B	Gemma 7B	RWKV6 7B
Method	AdaLoRA	AdaLoRA	AdaLoRA	AdaLoRA
Mix-Precision	TF32+BF16	TF32+BF16	TF32+BF16	TF32+BF16
Epoch	1	1	1	1
Batch Size	128	128	128	128
Max Length	512	512	512	512
Weight Decay	0	0	0	0
Warm-up Ratio	0.03	0.03	0.03	0.03
Learning Rate	2e-5	2e-5	2e-5	2e-5
Grad Clip	1.0	1.0	1.0	1.0
β_1	0.85	0.85	0.85	0.85
β_2	0.85	0.85	0.85	0.85
r_{init}	128	64	64	64
r_{target}	128	64	64	64
t_{init}	100	100	100	100
t_{final}	600	600	600	600

Table 5: Hyperparameters of evaluation for with AdaLoRA on different models for GSM-8K and MATH. ML denotes Max Length.

Model	Llama 2 7B	Mistral 7B	Gemma 7B	RWKV6 7B
Method	AdaLoRA	AdaLoRA	AdaLoRA	AdaLoRA
Precision	BF16	BF16	BF16	FP32
Sampling	False	False	False	False
Top-P	1.0	1.0	1.0	1.0
ML for GSM-8K	1024	1024	1024	1024
ML for MATH	2048	2048	2048	2048
ML for HumanEval	2048	2048	2048	2048