SINGULAR VALUE ADAPTATION FOR PARAMETER-EFFICIENT FINE TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Parameter-Efficient Fine-Tuning (PEFT) has become a crucial approach in handling the growing complexity of large models and vast datasets across multiple fields such as Computer Vision or Natural Language Processing. Among the most promising of these methods are Low-Rank Adaptation (LoRA) and its derivatives, which fine-tune a pre-trained weight matrix $\mathbf{W}$ by introducing a low-rank update matrix $\mathbf{\Delta W}$. While these approaches have demonstrated strong empirical performance, they remain largely heuristic, with little theoretical grounding to explain their behavior or guide the design of $\mathbf{\Delta W}$ for different objectives. This lack of theoretical insight limits our understanding of when these methods are most effective and how they can be systematically improved. In this paper, we propose a theoretical framework for analyzing and designing LoRA-based methods, with a focus on the formulation of $\mathbf{\Delta W}$. By establishing a deeper understanding of the interplay between $\mathbf{W}$ and $\mathbf{\Delta W}$, we aim to enable more efficient and targeted fine-tuning strategies, opening the door to novel variants that strike an optimal balance between performance and efficiency. Our proposed method - **Si**ngular **V**alue **A**daptation - uses insights from our theoretical framework to incorporate inductive biases on the formulation of $\mathbf{\Delta W}$, leading to a PEFT method that is up to 50$\times$ more parameter efficient that LoRA, while achieving comparable or better performance across various vision and language tasks.

1 INTRODUCTION

Large pre-trained neural networks have become indispensable across a wide array of domains, including natural language processing and computer vision, but fine-tuning them efficiently for specialized downstream tasks remains a significant challenge. The sheer scale of modern models makes fine-tuning all parameters computationally expensive and often impractical. To address this, various Parameter-Efficient Fine-Tuning (PEFT) methods have been developed, aiming to optimize model adaptation while minimizing computational overhead. PEFT techniques, such as Adapters, Prompts and Prefixes, Low-Rank Adaptation (LoRA), and their numerous variants, have gained popularity due to their ability to fine-tune models by only modifying a small number of parameters, making the process more efficient. These methods have been successfully deployed across various applications, demonstrating impressive performance gains while significantly reducing resource requirements. LoRA and its derivatives have in particular become widely prevelant, owing to the fact that these methods give comparable or improved performance over other PEFT methods, and do not incur any additional computational costs over the base model during inference.

Despite their success, there remains a lack of clarity regarding the underlying mechanisms that make LoRA and its derivatives effective. Most formulations of the low-rank update matrix are largely heuristic, with no studies on how different formulations affect the final merged matrix at inference. Our analysis on the original pretrained and and adapted weight matrices of a ViT-Base model finds that they do not significantly differ in terms of their rank (see table 1). Our findings reveal that most pre-trained matrices are in fact full-rank or near-full-rank, all. Instead, in this paper, we hypothesize that the effectiveness of LoRA based techniques may be driven by changes in the "effective rank" (Roy & Vetterli, 2007) of the model's weight matrices during fine-tuning. In table 1, we show initial observations supporting this hypothesis, where increases in the effective rank correlate with improved model performance, as evidenced by our sample results (Fig 1 (left)).

Building on this observation, we propose a novel method explicitly designed to maximize the increase in effective rank under the constraint of few trainable parameters.

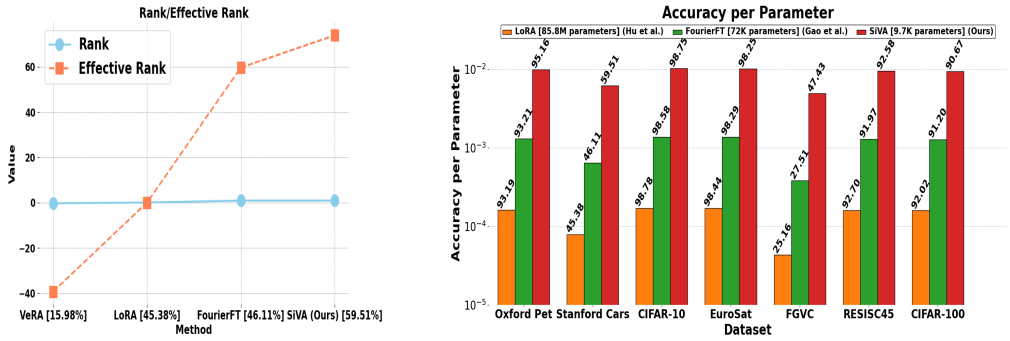

Figure 1: *(Left)* Plot of change in rank and effective rank of various PEFT methods, relative to an average baseline rank of 766.92 and an average baseline effective rank of 520.03 obtained from pre-trained ViT-Base model Query and Value matrices. Accuracy over Base model is computed using a linear probe (LP). The x-axis lists method names along with their respective accuracies on Stanford Cars dataset. *(Right)* Plot of accuracy per parameter for various PEFT methods, based on experiments conducted across 7 datasets using ViT-Base architecture. Accuracy for each method is displayed above the corresponding bar, number of trainable parameters for each method is provided in legend. Our PEFT method, SiVA, achieves significant parameter reduction while maintaining high accuracy, consistently delivering best accuracy per parameter performance.

Method	Accuracy	Parameters	Rank	Effective Rank
Base + LP	25.76	-	766.92	520.03
LoRA	45.38	581K	766.97 (+0.05)	520.06 (+0.03)
FourierFT	46.11	72K	767.85 (+0.90)	579.61 (+59.6)
VeRA	15.98	48K	767.26 (+0.34)	480.84 (-39.2)
SiVA (Ours)	59.51	9.7K	767.85 (+0.93)	593.86 (+73.8)

Table 1: Table depicting the accuracy over the classification task and the effective rank of the learned adapters on the Stanford Cars dataset over the ViT-B16 model. This illustrates that while the rank of the adapters are approximately equal across all methods, the accuracy seems to significantly correlate with its effective rank. Our method has the highest effective rank amongst the SoTA methods presented above as well as an accuracy score $\approx 13.5\%$ above them. The values in the brackets indicate the increase in rank/effective rank over the base model.

Our method, **Si**ngular **V**alue **A**daptation (**SiVA**), is grounded in theoretical insights about the relationships between the structure of the update matrix, its constraints on the number of trainable parameters, and its impact on the effective rank on the final matrix post training. By focusing on increasing the effective rank while maintaining a minimal parameter footprint, **SiVA** leverages the strengths of low-rank adaptation techniques while introducing new mechanisms to optimize effective rank growth. The contribution to the final performance by each trained parameter of SiVA - measured as *performance per parameter (PPP)* - is significantly higher compared to existing methods, even those that aggressively attempt to minimize the parameters via heuristic formulations of the LoRA update matrix (Kopiczko et al., 2024; Gao et al., 2024). A visualization of the PPP for different methods on an image classification task is shown in Fig 1(right), with additional visualizations shown later with other experimental results.

(1) The first component of our overarching theory makes use of the observation that the effective rank of a matrix is solely a function of its singular values. Based on this insight, we derive a relation for the contribution made by each individual singular value update to the effective rank of the merged matrix. (2) Next, we realize that to achieve parameter efficiency, we only need to update (train) a subset of all singular values in each weight matrix of the base model. This leads us to derive that the smaller singular values play a greater role in maximizing the effective rank of the final matrix, and thus they are the ones that should be tuned, while the larger singular values can be left unchanged. (3) Finally, we prove the intuitive result that the left and right singular vectors of the adapted matrix should be aligned with those of the pre-trained weight matrix to maximize the effective rank (of the adapted matrix). Overall, these three components reveal a simple formulation of the low-rank update matrix - the update matrix can be written as the composition of three matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{S}_{\mathbf{\Delta W}}$, as $\mathbf{\Delta W} = \mathbf{U}\mathbf{S}_{\mathbf{\Delta W}}\mathbf{V}^{\mathbf{T}}$, where $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors of a pre-trained weight matrix, the lower elements of the diagonal matrix $\mathbf{S}_{\mathbf{\Delta W}}$ are learned using Gradient Descent, and the other elements of $\mathbf{S}_{\mathbf{\Delta W}}$ are set to zero. (4) To conclude our theoretical insights, we show that this formulation can achieve the minimum with zero loss in the least squares regression problem when all singular values are trainable.

Our overall contributions can be summarized as follows: (i) We show that the performance of transformer models on downstream tasks is correlated to the effective ranks of the query and value matri-

ces, an insight that was previously unknown; (ii) We propose a specialized structure for the composition of the update matrix, and theoretically show that the proposed structure maximizes the effective rank of the merged weights post-training. We propose a simple, yet highly performant LoRA-based approach inspired by our theoretical insights; (iii) We evaluate the performance of the proposed approach across several tasks spanning both Computer Vision and Natural Language Processing, and show that our approach, despite having a simple formulation, outperforms state-of-the-art methods in terms of performance-per-parameter by large margins. In terms of absolute performance across different metrics, our approach achieves results comparable to or better than existing PEFT methods.

2 Related Work

LoRA and Variants. LoRA-based PEFT methods solve the computational overhead of fine-tuning large models by modelling the change in parameters as a low-rank matrix. Various decompositions of the low-rank matrix have been proposed, leading to an entire family of LoRA derivatives (Kopiczko et al., 2024; Liu et al., 2024; Gao et al., 2024; Aghajanyan et al., 2021; Karimi Mahabadi et al., 2021; Edalati et al.; Liao et al., 2023; He et al., 2023). The original LoRA formulation (Hu et al., 2022) decomposed all weight matrices as a product of two rectangular learnable matrices of specified rank. Dynamic-rank LoRA derivatives (Zhang et al., 2023; 2024; Ding et al., 2023; Valipour et al., 2023; Haobo et al., 2024) methods further refine this approach by using adaptive ranks for different layers. More recent works propose the use of pseudo-random vectors or matrices to achieve aggressive parameter compression. NOLA (Koohpayegani et al., 2024) learns the coefficient of linear combination of pseudo-random matrices, while FourierFT (Gao et al., 2024) samples a random spectral basis and learns the sparse spectral coefficients , using an Inverse Discrete Fourier Transform to get the weight update matrix. In similar spirit, VeRA (Kopiczko et al., 2024) samples two random matrices, and scales them by learnable factors before multiplying them to obtain the update matrix. These methods are largely heuristic, without any underlying common principle that guides the formulation of the weight update matrix.

Other PEFT Methods. Adapter Tuning methods (Pfeiffer et al., 2021; Houlsby et al., 2019b; Lei et al., 2023; He et al.; Zhu et al., 2021) introduce task-specific parameters through small layers (adapters) inserted within a pre-trained model. Prompt/Prefix Tuning methods (Li et al., 2023; Liu et al.; Zhang et al.; Zhu & Tan; Wu et al., 2022; Ma et al., 2022; Lester & Constant; Liu et al., 2023) prepend learnable vectors to the inputs of the model (prompts) or individual intermediate layers (prefixes). Some other methods include BitFit (Zaken et al., 2022) which only tunes the model biases and IA3 (Liu et al., 2022) which introduces additional parameters in the Self-Attention module. All these methods increase the complexity of the base model, leading to increased inference times and a requirement for additional space. LoRA derivatives do not suffer from this issue as the newly learned parameters can be directly added to the parameters of the base model post training.

3 SiVA: Formulation and Methodology

3.1 Notations and Preliminaries

Our work is based on the hypothesis that the performance of the adapted model is proportional to the increase in effective rank caused by adding the learned matrix $\mathbf{\Delta W}$ to the pre-trained weight matrix $\mathbf{W}$. Crucially, we want to maximize the effective rank while keeping the number of parameters in $\mathbf{\Delta W}$ minimum. Since adapter-based methods are usually applied to the square-shaped query and value matrices in transformers, we restrict our formulation to square matrices. Formally, the **effective rank** (Roy & Vetterli, 2007) of a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is given by

$$\texttt{erank}(\mathbf{A}) = e^{\mathcal{H}_{\mathbf{A}}} = e^{\mathcal{H}_{\mathbf{\Sigma_A}}},$$

where $\Sigma_{\mathbf{A}}$ represents the diagonal matrix of singular values $(\sigma_1, \sigma_2, ..., \sigma_n)$ of $\mathbf{A}$. Here $\mathcal{H}_{\mathbf{\Sigma_A}} := -\sum_{i=1}^{n} p_i log(p_i)$, where p_i is the i^{th} normalized singular value computed as $p_i = \frac{\sigma_i}{\sum_{j=1}^{n} \sigma_k}$. We refer to $\mathcal{H}_{\mathbf{A}}$ as the *entropy* of matrix $\mathbf{A}$. Furthermore, for a discrete random variable χ that obeys a distribution $\mathcal{P}$ with finite support over n states $(\bigcup_{i=1}^{n} \chi_i)$, we define $H(\mathcal{P}) = -\sum_{i=1}^{n} p_i \log(p_i)$, where $p_i = \texttt{Prob}(\chi = \chi_i)$. In this work, we restrict ourselves to dealing with discrete random variables alone.

We posit that after fine-tuning the adapters attached to the query and value matrices, the effective rank of the combined matrices should be greater than the original matrix to increase performance on the downstream task. In other words, if $\mathbf{W}$ represents the original weight matrix of an adapted query

or value, and $\mathbf{\Delta W}$ represents the weight matrix provided by the adapter module *post fine-tuning on a downstream task*, then $\texttt{erank}(\mathbf{W} + \mathbf{\Delta W}) > \texttt{erank}(\mathbf{W})$. We observe that the effective rank of the resulting matrix depends solely on its singular values. Based on this observation, we now present our theory to analyze the conditions under which $\mathbf{W} + \mathbf{\Delta W}$ has the highest effective rank over $\mathbf{W}$ under certain constraints on $\mathbf{\Delta W}$.

3.2 THEORETICAL FORMULATION

Since the effective rank increases monotonically with the entropy $\mathcal{H}$, we shift our focus towards the analysis of $\mathcal{H}$. We first derive an expression for computing the change in entropy of the combined matrix over the original matrix, i.e. $\Delta\mathcal{H} = \mathcal{H}_{\mathbf{W}+\mathbf{\Delta W}} - \mathcal{H}_{\mathbf{W}}$. Using this expression, we proceed to show that for maximizing $\mathbf{\Delta}\mathcal{H}$, the change in the ith singular value of $\mathbf{W}$ should be proportional to its own magnitude.

The next part of our analysis uses this observation to find the ideal singular values to learn given that we wish to constrain the number of parameters (here, singular values) that are trained. We proceed to show that if all singular values are not freely tunable, it is optimal to tune the lower singular values of $\mathbf{W}$ using $\mathbf{\Delta W}$. Additionally, we further show that for two different $\mathbf{\Delta W}$ matrices that have the same norm and the same number of tunable singular values, the one that has its singular vectors aligned with the singular values of $\mathbf{W}$ results in the maximum increase in the effective rank.

We then show that this formulation - training only the lower singular values of $\mathbf{W}$ by adding an appropriately crafted $\mathbf{\Delta W}$ - achieves an optimal solution for the Linear Regression Problem, and the proceed to adapt this method for fine-tuning transformers.

Lemma 1 (Change in Entropy)**.** *Given a random variable χ and any two distributions $\mathcal{P},\ \mathcal{Q}$ over $(\bigcup_{i=1}^{n} \chi_i)$, define $\delta p_i := \mathcal{P}(\chi = \chi_i) - \mathcal{Q}(\chi = \chi_i)$, and assume that such that $|\delta p_i| << 1,\ \forall 1 \leq i \leq n$. Then the change in entropy $\Delta H_{q,p} := H(q) - H(p)$ is given by: $\Delta H_{q,p} = -\sum_{i=1}^{n} \delta p_i (\log p_i)$.*

Our primary result investigates the optimal perturbations introduced by the adapter, $\mathbf{\Delta W}$, to the pre-trained weight matrix, $\mathbf{W}$. The proof outline begins by deriving a simplified representation of the singular values of the adapted weight matrix, $\mathbf{W} + \mathbf{\Delta W}$, utilizing a first-order Taylor expansion around the normalized singular values. A first-order approximation is sufficient in this case, as the adjustment to the i^{th} singular value, $\delta\sigma_i$, is generally much smaller in magnitude compared to the corresponding singular value of $\mathbf{W}$.

Subsequently, we employ our central technical Lemma 1 to analyze the distributions induced by the normalized singular values of $\mathbf{W} + \mathbf{\Delta W}$ and $\mathbf{W}$, respectively. Maximizing the relevant entropy reduces the problem to optimizing an objective function over the free variables $\delta\sigma_i$. To account for regularization constraints on $\mathbf{\Delta W}$, a common practice in the PEFT literature (Hu et al., 2022), we incorporate a constraint on the Frobenius norm of $\mathbf{\Delta W}$. The optimal adjustments for this constrained optimization are then derived using the method of Lagrange multipliers. The question of whether these optimal adjustments scale linearly with the magnitude of the corresponding singular values in $\mathbf{W}$ is resolved affirmatively in the following theorem.

Theorem 1 (Increase in Entropy under Singular Value Adjustment)**.** *Let $A \in \mathbb{R}^{n \times n}$ be a matrix with singular values $\sigma_i^A \geq 0$ for $i = 1, 2, \ldots, n$, and let $S_A = \sum_{i=1}^{n} \sigma_i^A$. Consider perturbations $\delta\sigma_i \in \mathbb{R}$ such that the singular values of $A + B$ become $\sigma_i^{A+B} = \sigma_i^A + \delta\sigma_i$, for an arbitrary matrix $B \in \mathbb{R}^{n \times n}$. Under the constraint $\|B\|_F^2 = \sum_{i=1}^{n} (\delta\sigma_i)^2 \leq C$, the maximum possible change in entropy ΔH is:*

$$\Delta H_{\max} = \sqrt{C} \cdot \sqrt{\sum_{i=1}^{n} c_i^2}, \tag{1}$$

where $c_i = -\frac{\left(\log\left(\frac{\sigma_i^A}{S_A}\right) + \mathcal{H}_A\right)}{S_A}$. The optimal adjustments $\delta\sigma_i$ are given by:

$$\delta\sigma_i = c_i \cdot \frac{\sqrt{C}}{\sqrt{\sum_{j=1}^{n} c_j^2}}. \tag{2}$$

All proofs are provided in the Appendix. Even if one optimizes the singular values of $\mathbf{W} + \mathbf{\Delta W}$ to follow the structure outlined in Theorem 1, a critical question remains: how does this optimization

contribute to parameter reduction when one might need to train n such singular values? This could lead to the training of $\mathbf{\Delta W} \in \mathbb{R}^{n \times n}$, resulting in as many parameters as the original pre-trained matrix, making it inefficient from a parameter-efficiency standpoint.

To address this, we introduce Theorem 2. It shows that if updates are restricted to only some k of the singular values of $\mathbf{W}$, while the remaining $n-k$ singular values remain unchanged, it is optimal under mild conditions to update the k smallest singular values. This can be intuitively expected in the following sense: the entropy of a discrete random variable is maximized when the probabilities of it taking different values are equi-probable. Formally, this result is derived by tweaking Equation 1 and utilizing the properties of the constants c_i, as detailed in Theorem 1, which are further explored in the Appendix.

Theorem 2 (Sparse, Optimal Modification of Singular Values to Maximize Entropy Increase)**.** *Let $A, B \in \mathbb{R}^{n \times n}$ be two matrices, with A being fixed. Suppose we are allowed to perturb at most k singular values of $A+B$ (i.e., at most k of the $\delta\sigma_i$ are non-zero), using a matrix $B \in \mathbb{R}^{n \times n}$ under the constraint $\|B\|_F^2 = \sum_{j=1}^{k} (\sigma_j^B)^2 \leq C$. To maximize the increase in entropy $\Delta H = \mathcal{H}_{A+B} - \mathcal{H}_A$, it is optimal to modify the k-smallest singular values of A.*

Theorem 2 offers a significant reduction in the number of parameters that need to be trained, depending on the desired downstream task performance (these requirements are encoded in the values of k and C). However, this still involves learning parameters for the orthonormal matrices $U_{\mathbf{\Delta W}}$ and $V_{\mathbf{\Delta W}}$, which are part of the singular value decomposition (SVD) of $\mathbf{\Delta W}$. To further reduce the number of parameters, we focus on minimizing the parameters to be trained in these orthonormal matrices. It is easy to observe that no additional parameters would be required to learn these matrices if, after training, the singular vectors of $\mathbf{\Delta W}$ align with those of $\mathbf{W}$, irrespective of the initialization of the entries of $\mathbf{\Delta W}$. In fact, under mild conditions, this is precisely what we prove in Theorem 3. This result is essential in justifying our formulation, where we explicitly constrain the orthonormal matrices in the SVD of $\mathbf{\Delta W}$ to be identical to those of $\mathbf{W}$. This provides the second major reduction in parameters compared to other PEFT methods, as we ultimately only train scalar values representing adjustments to the pre-trained weights $\mathbf{W}$. The proof primarily follows a greedy argument, relying on Lemma 1 and a result from first-order perturbation theory (Stewart, 1998).

Learning coefficients for linear combinations has been explored in prior work (Koohpayegani et al., 2024; Kopiczko et al., 2024; Gao et al., 2024), where adapter weights are formulated as linear combinations of matrices from a basis of randomly chosen matrices. In contrast, our approach utilizes fixed deterministic matrices derived from the pre-trained model. An additional advantage of our method is that it remains invariant to the implementation of random number generators, unlike these other methods, and it does not require storing random seeds post-training. Furthermore, our approach removes a significant constraint required by previous methods: we do not require the adapter matrices to span the same random basis at every layer. Instead, at each layer the singular vectors of the adapter matrix are aligned with the singular vectors of the corresponding weight matrix, enhancing the expressivity of our method while avoiding cross-talk between weight matrices across layers.

Theorem 3 (Alignment of Singular Vectors w.r.t Pretrained Weights)**.** *Let $A \in \mathbb{R}^{n \times n}$ be a matrix with singular values σ_i^A arranged in descending order ($\sigma_1^A \geq \sigma_2^A \geq \cdots \geq \sigma_n^A \geq 0$) and corresponding left and right singular vectors $\mathbf{u}_i^A$ and $\mathbf{v}_i^A$. Let $B \in \mathbb{R}^{n \times n}$ be a fixed matrix with exactly k nonzero singular values σ_j^B (with $\sigma_1^B \geq \sigma_2^B \geq \cdots \geq \sigma_k^B > 0$) and corresponding singular vectors $\mathbf{u}_j^B$ and $\mathbf{v}_j^B$. Under the constraint $\|B\|_F^2 = \sum_{j=1}^{k} (\sigma_j^B)^2 = C$, the maximum increase in entropy $\Delta H = \mathcal{H}_{A+B} - \mathcal{H}_A$ is achieved when the following happen, in order:*

1. *First, the singular vectors of B corresponding to its largest singular value are aligned with the singular vectors of A corresponding to its smallest singular value; specifically, $\mathbf{u}_j^B = \mathbf{u}_{n-j+1}^A$ and $\mathbf{v}_j^B = \mathbf{v}_{n-j+1}^A$ for $j = 1, 2, \ldots, k$.*
2. *Since the largest singular value of B is now aligned, ΔH is further maximized by aligning the next largest singular value of B with the next smallest singular vector of A, and so on recursively, for all k singular values of B.*

Therefore, the largest increase in entropy is achieved by aligning the singular values of B in decreasing order with the singular vectors of A in increasing order of their indices.

Theorems 1–3 provide a strong theoretical foundation for the PEFT formulation that we introduce as **SiVA**. However, to rigorously establish that **SiVA** achieves optimal performance, we need to connect this formulation directly to the loss function used during training. It is important to note that we present experiments for **SiVA** on large, deep networks, such as ViT-B16 (small/large), which incorporate multiple non-linearities and attention heads, thereby inducing complex inductive biases into the learned weights. Fully characterizing the trajectory of the solutions derived by **SiVA** under these conditions is highly non-trivial and left as future work.

Instead, we show that in a simpler Linear Regression setting, there exist weight matrices $\mathbf{\Delta W}$ whose singular vectors align with those of $\mathbf{W}$ (**SiVA** form) and are within the set of weight matrices that minimize the loss function. The proof uses a standard proof-by-contradiction approach: we begin by assuming that no such solution exists, and then construct an optimal solution of the **SiVA** form for the regression task. By demonstrating the existence of this optimal solution, we invalidate the original assumption, proving that a **SiVA**-form solution is optimal for the linear regression task. The proof is constructive and can be derived using standard derivative-based optimization techniques.

Theorem 4 (**SiVA** Style Solutions Lie in Set of Minimizers for Linear Regression)**.** *Let $A \in \mathbb{R}^{m\times n}$ be a full-rank matrix with singular value decomposition $A = U_A \Sigma_A V_A^\top$, where $U_A \in \mathbb{R}^{m\times n}$ and $V_A \in \mathbb{R}^{n\times n}$ are orthogonal matrices, and $\Sigma_A \in \mathbb{R}^{n\times n}$ is a diagonal matrix with positive entries $\sigma_{A1}, \sigma_{A2}, \ldots, \sigma_{An}$. Let $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ be given vectors. Define $B = U_A S V_A^\top$, where $S \in \mathbb{R}^{n\times n}$ is a diagonal matrix with entries $\sigma_1, \sigma_2, \ldots, \sigma_n$. Consider the Mean Squared Error:*

$$L(\sigma_1, \ldots, \sigma_n) = \|(A + B)x - y\|^2 .$$

Then, $L(\sigma_1, \ldots, \sigma_n)$ is minimized by choosing

$$\sigma_i = \frac{u_i^\top y}{v_i^\top x} - \sigma_{Ai},$$

for each i such that $v_i^\top x \neq 0$, where u_i and v_i are the i-th columns of U_A and V_A, respectively.

3.3 Method

This section outlines the implementation of our method. We use insights from our theoretical observations to develop a PEFT method that is conceptually simple, aggressively parameter-efficient, and highly performant. Our approach directly draws inspiration from our theoretical framework, and is based on the following hypothesis - *Modification of a small subset of singular values of the pre-trained model weights is sufficient to achieve high performance, while simultaneously limiting the number of parameters to be trained.* We formulate $\mathbf{\Delta W}$ such that it selectively updates the lower singular values of a pre-trained weight matrix $\mathbf{W}$ when added to it. The pseudocode outlining the various components of SiVA is presented in Algorithm 1.

Formally, let $\mathbf{W}$ be the weight matrix to be adapted to a downstream task by adding the update matrix $\mathbf{\Delta W}$. We compose $\mathbf{\Delta W}$ as product of three matrices $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V^T}$, where $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors of $\mathbf{W}$ respectively (Theorem 4). Note that this initialization of $\mathbf{U}$ and $\mathbf{V}$ is a one-time operation, performed only when attaching the **SiVA** module to the base layer. $\mathbf{S}$ is a diagonal matrix whose values are optimized using Gradient Descent.

To achieve parameter efficiency, we only train k diagonal elements of $\mathbf{S}$, corresponding to the singular values of $\mathbf{W}$ that need to be modified. The remaining singular values are frozen, along with $\mathbf{U}$ and $\mathbf{V^T}$. Only the lowest singular values are chosen for training (Theorem 3). Analogous to the observation made by AdaLoRA (Zhang et al., 2023) about the suboptimal choice of having the same rank for each update matrix, maintaining a constant number of $\mathbf{k}$ per layer can also be suboptimal. Therefore, we determine $\mathbf{k}$ in each layer based on the variance captured by the corresponding eigenvalues. We pick the singular values corresponding to the eigenvalues that capture the bottom $(1 - perc)\%$ of the variance, where $perc$ is a hyperparameter. The pseudocode for computing k for a given value of $perc$ is presented in the *get_values_and_vectors* function in Algorithm 1.

4 Experiments and Results

We evaluate the performance of SiVA across computer vision (CV) and natural language processing (NLP) datasets. For CV, SiVA fine-tunes **(1)** vision transformers (ViT) (Dosovitskiy et al., 2021) in Base and Large variants for image classification. For NLP, SiVA is applied to **(2)** RoBERTa-Large (Liu et al., 2020) for natural language understanding on the GLUE (Wang et al., 2018) benchmark,

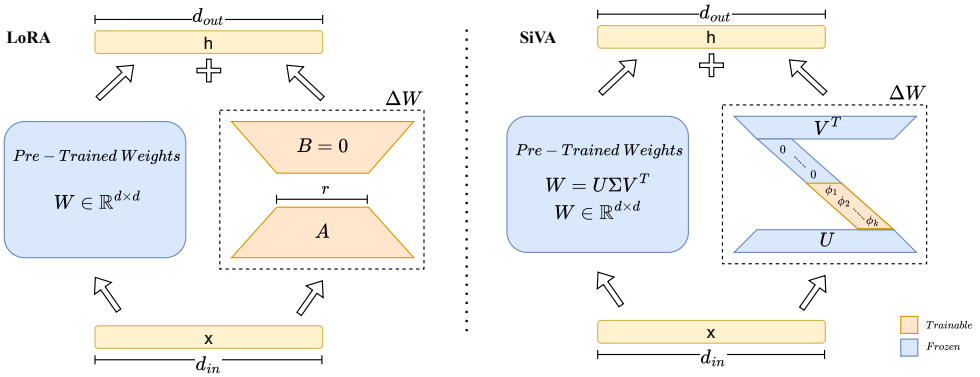

Figure 2: Visualization of LoRA (left) against SiVA (right). Compared to LoRA, SiVA has significantly fewer trainable components. Additionally, all frozen components in SiVA are derived from the base matrix itself.

Algorithm 2 PyTorch-style Pseudocode for our method, SiVA

```
class SiVA(nn.Module):
    def __init__(
        self,
        perc: float, # only train the singular values corresponding to the eigenvalues
        # capturing bottom (1 - perc)% variance
        base_layer: nn.Module # pre-trained layer,
    ):
        # definitions
        self.perc = perc
        self.base_layer = base_layer

        # Get the the vector of lower singular values,
        # and corresponding left and right singular vectors
        self.U, S, self.V = self.get_values_and_vectors(base_layer.weight)
        self.S = nn.Parameter(torch.zeros_like(S))

    def get_values_and_vectors(self, matrix):
        U, S, V = torch.svd(matrix)
        # Calculate the total variance (sum of squared singular values)
        total_variance = (S**2).sum()

        # Calculate the cumulative sum of squared singular values
        cumulative_variance = torch.cumsum(S**2, -1)

        # Find the number of singular values that capture perc% of the variance
        num_frozen_singular_values = torch.searchsorted(
            cumulative_variance,
            self.perc * total_variance
        ) + 1

        num_trainable_singular_values = S.shape[0] - num_singular_values

        reduced_U = U[:, -num_trainable_singular_values:]
        reduced_S = S[-num_trainable_singular_values:]
        reduced_V = V[:, -num_trainable_singular_values:]
        return reduced_U, reduced_S, reduced_V

    def forward(self, x):
        result = self.base_layer(x)
        W = self.U @ self.S.diag() @ self.V.t()
        result += F.linear(x, W)
        return result
```

(3) GPT-2 (Medium) (Radford et al., 2019) for natural language generation on the E2E dataset (Wang et al., 2023), and **(4)** LLaMA2-7B (Touvron et al., 2023) for instruction tuning. In each case, along with standard performance metrics for the task, we also report the *performance per parameter*. This quantity indicates how much each trained parameter contributes to the final performance on average. Additionally, we study the effects of: **(1)** using singular vectors derived from original weights versus random matrices, and **(2)** using upper versus lower singular values.

4.1 Image Classification

Models and Datasets. SiVA is evaluated on the image classification task using the Vision Transformer (ViT) (Dosovitskiy et al., 2021), in both Base and Large variants. The models are pre-trained on the ImageNet-21K dataset (Ridnik et al.), and fine-tuned on OxfordPets (Parkhi et al., 2012)(37 classes), CIFAR10 (10 classes) (Krizhevsky, 2009), EuroSAT (10 classes) (Helber et al., 2019), RESISC45 (45 classes) (Cheng et al., 2017), StanfordCars (196 classes) (Krause et al., 2013), FGVC (100 classes) (Maji et al.), and CIFAR100 (100 classes) (Krizhevsky, 2009). We compare SiVA against two established parameter-efficient fine-tuning (PEFT) methods: LoRA (Hu et al., 2022)

and FourierFT (Gao et al., 2024). Additionally, we also show results for training a single Linear Layer on top of the base model (**LP**) and full fine-tuning of the base model (**FF**). We reuse baseline numbers from FourierFT to ensure fairness.

Implementation Details. SiVA is evaluated against Full Fine-tuning (FF), Linear Probing (LP, fine-tuning the classification head only), LoRA (Hu et al., 2022), and FourierFT (Gao et al., 2024). For LoRA, FourierFT and SiVA , we fine-tune the query and value matrices in ViT. We report the results for using using $r = 16$ for LoRA, $n = \{3000\}$ for FourierFT, and $perc = 0.95$ for SiVA . Learning rates and weight decay are tuned, with a maximum of 10 training epochs. Hyperparameters are provided in Table 7 in the Appendix.

Model	Method	# Trainable Parameters	OxfordPets	StanfordCars	CIFAR10	EuroSAT	FGVC	RESISC45	CIFAR100	Avg.
ViT-Base	LP	-	90.28$_{\pm 0.43}$	25.76$_{\pm 0.28}$	96.41$_{\pm 0.02}$	88.72$_{\pm 0.13}$	17.44$_{\pm 0.43}$	74.22$_{\pm 0.10}$	84.28$_{\pm 0.11}$	68.16
	FF	85.8M	93.14$_{\pm 0.40}$	79.78$_{\pm 1.15}$	98.92$_{\pm 0.05}$	99.05$_{\pm 0.09}$	54.84$_{\pm 1.23}$	96.13$_{\pm 0.13}$	92.38$_{\pm 0.13}$	87.75
	LoRA (Hu et al., 2022)	581K	93.19$_{\pm 0.36}$	45.38$_{\pm 0.41}$	**98.78**$_{\pm 0.05}$	**98.44**$_{\pm 0.15}$	25.16$_{\pm 0.16}$	**92.70**$_{\pm 0.18}$	**92.02**$_{\pm 0.12}$	77.95
	FourierFT (Gao et al., 2024)	72K	93.21$_{\pm 0.26}$	46.11$_{\pm 0.24}$	98.58$_{\pm 0.07}$	98.29$_{\pm 0.04}$	27.51$_{\pm 0.64}$	91.97$_{\pm 0.31}$	91.20$_{\pm 0.14}$	78.12
	SiVA	**9.7K**	**95.16**$_{\pm 0.53}$	**59.51**$_{\pm 0.4}$	98.75$_{\pm 0.07}$	98.25$_{\pm 0.04}$	**47.43**$_{\pm 1.7}$	92.58$_{\pm 0.35}$	90.67$_{\pm 0.10}$	**83.19**
ViT-Large	LP	-	91.11$_{\pm 0.30}$	37.91$_{\pm 0.27}$	97.78$_{\pm 0.04}$	92.64$_{\pm 0.08}$	24.62$_{\pm 0.24}$	82.02$_{\pm 0.11}$	84.28$_{\pm 0.11}$	72.91
	FF	303.3M	94.43$_{\pm 0.56}$	88.90$_{\pm 0.26}$	99.15$_{\pm 0.05}$	99.04$_{\pm 0.08}$	68.25$_{\pm 1.63}$	96.43$_{\pm 0.07}$	93.58$_{\pm 0.19}$	91.4
	LoRA (Hu et al., 2022)	1.57M	94.82$_{\pm 0.09}$	73.25$_{\pm 0.36}$	99.13$_{\pm 0.03}$	98.63$_{\pm 0.07}$	42.32$_{\pm 0.98}$	**94.71**$_{\pm 0.25}$	**94.87**$_{\pm 0.10}$	85.39
	FourierFT (Gao et al., 2024)	144K	94.46$_{\pm 0.28}$	69.56$_{\pm 0.30}$	99.10$_{\pm 0.04}$	**98.65**$_{\pm 0.09}$	39.92$_{\pm 0.68}$	93.86$_{\pm 0.14}$	93.31$_{\pm 0.09}$	84.12
	SiVA	**30.3K**	**95.98**$_{\pm 0.2}$	**79.61**$_{\pm 0.52}$	**99.16**$_{\pm 0.07}$	**98.65**$_{\pm 0.1}$	**54.19** $_{\pm 1.9}$	94.52$_{\pm 0.2}$	92.39$_{\pm 0.3}$	**87.79**

Table 2: Fine-tuning results with ViT Base and Large models on different image classification datasets. We report the accuracy (%) after 10 epochs. Avg. represents the average accuracy of each method on all datasets. We show the best performance across PEFT methods in **bold**, and underline the next best PEFT method.

Results. Table 2 presents the results across seven image classification datasets. All three adapter-based methods significantly outperform Linear Probing. SiVA achieves comparable or better performance with **two orders of magnitude fewer** parameters compared to LoRA and an order of magnitude fewer parameters compared to FourierFT, which is a method that attempts to aggressively reduce number of parameters over LoRA. We also outperform other methods on performance per parameter metric by a significant margin (see Fig 1), achieving state-of-the-art results on all benchmark datasets.

4.2 Natural Language Understanding

Models and Datasets. SiVA is evaluated on the GLUE benchmark (Wang et al., 2018), covering various natural language understanding tasks such as sentence classification, paraphrase detection, and natural language inference. We fine-tune the RoBERTa-Large model (Liu et al., 2020) for this evaluation. We compare our approach to multiple other parameter-efficient fine-tuning approaches: **Full Fine-tuning (FF)**, where all parameters are updated during fine-tuning, starting from pre-trained weights and biases; **Bitfit** (Zaken et al., 2022), where the biases are tuned while keeping all other parameters fixed; Three variants of **Adapter Tuning** Houlsby et al. (2019a), which introduces two-layer adapters between frozen transformer layers; **LoRA** (Hu et al., 2022), which formulates parameter updates as a product of two low-rank matrices; **DyLoRA** (Valipour et al., 2023) and **AdaLoRA** (Zhang et al., 2023) both of which dynamically optimize the rank of LoRA matrices; **VeRA** (Kopiczko et al., 2024), which employs "scaling vectors" to adapt a pair of frozen random matrices shared between layers for weight updates, and finally, **FourierFT** (Gao et al., 2024), which leverages the Inverse Fourier Transform to learn parameters in the frequency domain and translate them into the weight space.

Implementation Details. We train singular values corresponding to the bottom 5% variance of weight matrices (i.e., $perc = 0.95$ across 24 layers). For all six GLUE tasks, we tune the learning rates of the head and SiVA parameters. The fine-tuning setup is similar to Hu et al. (2022), targeting the query and value matrices in each transformer block, with full fine-tuning of the classification head. Hyperparameters are detailed in Table 9 in the Appendix.

Results. Table 3 summarizes the results. We present the mean over five random seeds with the best epoch selected. SiVA matches or surpasses baseline methods with significantly fewer trainable parameters, including Full Fine-tuning in some cases, such as for MRPC. We outperform other methods on performance per parameter metric by a large margin (See Fig 3).

Method	# Trainable Parameters	SST-2 (Acc.)	MRPC (Acc.)	CoLA (MCC)	QNLI (Acc.)	RTE (Acc.)	STS-B (PCC)	Avg.
FF	356M	96.4	90.9	68	94.7	86.6	92.4	88.2
Adpt^{P} (Pfeiffer et al., 2021)	3M	$96.1_{\pm 0.3}$	$90.2_{\pm 0.7}$	$68.3_{\pm 1.0}$	$94.8_{\pm 0.2}$	$83.8_{\pm 2.9}$	$92.1_{\pm 0.7}$	87.6
Adpt^{P} (Pfeiffer et al., 2021)	0.8M	$96.6_{\pm 0.2}$	$89.7_{\pm 1.2}$	$67.8_{\pm 2.5}$	$94.8_{\pm 0.3}$	$80.1_{\pm 2.9}$	$91.9_{\pm 0.4}$	86.8
Adpt^{H} (Houlsby et al., 2019b)	6M	$96.2_{\pm 0.3}$	$88.7_{\pm 2.9}$	$66.5_{\pm 4.4}$	$94.7_{\pm 0.2}$	$83.4_{\pm 1.1}$	$91.0_{\pm 1.7}$	86.8
Adpt^{H} (Houlsby et al., 2019b)	0.8M	$96.3_{\pm 0.5}$	$87.7_{\pm 1.7}$	$66.3_{\pm 2.0}$	$94.7_{\pm 0.2}$	$72.9_{\pm 2.9}$	$91.5_{\pm 0.5}$	84.9
LoRA (Hu et al., 2022)	0.8M	$\mathbf{96.2}_{\pm 0.5}$	$90.2_{\pm 1.0}$	$\underline{68.2}_{\pm 1.9}$	$\mathbf{94.8}_{\pm 0.3}$	$85.2_{\pm 1.1}$	$\mathbf{92.3}_{\pm 0.5}$	87.8
FourierFT (Gao et al., 2024)	$\underline{0.048\text{M}}$	$96.0_{\pm 0.2}$	$\underline{90.9}_{\pm 0.3}$	$67.1_{\pm 1.4}$	$\underline{94.4}_{\pm 0.4}$	$\mathbf{87.4}_{\pm 1.6}$	$91.9_{\pm 0.4}$	$\underline{88.0}$
VeRA (Kopiczko et al., 2024)	0.061M	$\underline{96.1}_{\pm 0.1}$	$90.9_{\pm 0.7}$	$68.0_{\pm 0.8}$	$\underline{94.4}_{\pm 0.2}$	$85.9_{\pm 0.7}$	$91.7_{\pm 0.8}$	87.8
SiVA	**0.023M**	$\mathbf{96.2}_{\pm 0.1}$	$\mathbf{91.4}_{\pm 0.4}$	$\mathbf{68.4}_{\pm 0.9}$	$94.2_{\pm 0.1}$	$\underline{87.1}_{\pm 0.1}$	$\underline{92.0}_{\pm 0.1}$	**88.22**

Table 3: Performance of various fine-tuning methods with RoBERTa Large on 6 tasks of the GLUE benchmark. We report the Matthew's correlation coefficient (MCC) for CoLA, Pearson correlation coefficient (PCC) for STS-B and accuracy (Acc.) for all the remaining tasks. We report the mean result of 5 runs with different seeds, each using different random seeds. The best results across PEFT methods for each dataset are shown in **bold**, and the second best results are underlined. Higher is better for all metrics except the number of trainable parameters, where lower is better.

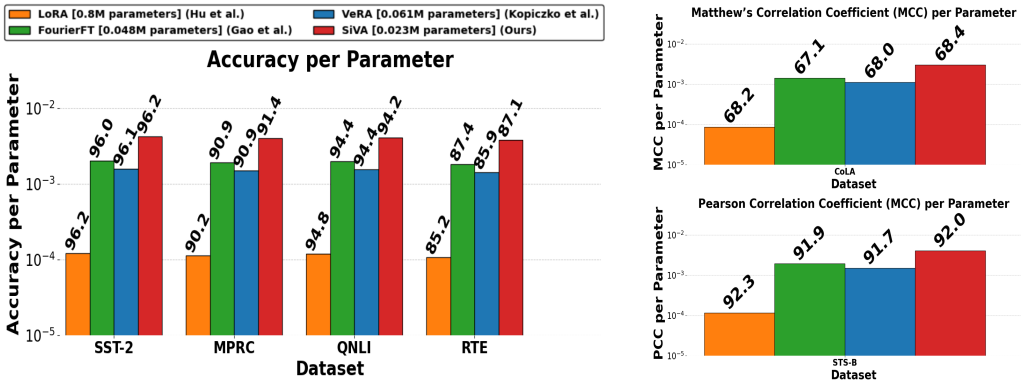

Figure 3: *(Left)* Accuracy per parameter for SST-2, MPRC,QNLI and RTE datasets. Accuracy for each method on each dataset is displayed above the corresponding bar, while number of trainable parameters for each method is provided in legend. *(Right)* Matthew's correlation coefficient (MCC)/ Pearson correlation coefficient (PCC) per parameter for CoLA/STS-B dataset. MCC/PCC for each method is displayed above corresponding bar, while number of trainable parameters for each method is provided in legend. SiVA demonstrates significantly better performance per parameter across the results.

4.3 Natural Language Generation

Models and Datasets. SiVA is evaluated on the E2E NLG task (Wang et al., 2023), using GPT-2 Medium (354M) model. The E2E dataset includes about 42,000 training samples and 4,600 samples each for validation and testing in the restaurant domain.

Implementation Details. We reuse baseline results from previous works, except for LoRA and SiVA , which are fine-tuned using a linear learning rate scheduler over five epochs. The batch size and learning rate are tuned, and the last epoch is selected for evaluation across three runs. Hyperparameters are detailed in Table 8 in the appendix.

Results. As shown in Table 4, SiVA achieves comparable or better performance than state-of-the-art methods across all metrics. SiVA does this with the lowest number of parameters, with order of magnitude fewer parameters compared to LoRA.

Method	# Trainable Parameters	BLEU	NIST	METEOR	ROUGE-L	CIDEr
FT	354.92M	68.2	8.62	46.2	71.0	2.47
Adpt^{L}	0.37M	66.3	8.41	45.0	69.8	2.40
Adpt^{L}	11.09M	68.9	8.71	46.1	71.3	2.47
Adpt^{H}	11.09M	67.3	8.5	46.0	70.7	2.44
LoRA	0.35M	68.9	8.76	$\underline{46.6}$	$\underline{71.5}$	**2.53**
FourierFT	$\underline{0.048\text{M}}$	$\underline{69.1}$	**8.82**	**47.0**	**71.8**	$\underline{2.51}$
SiVA	**0.044M**	**69.7**	$\underline{8.81}$	46.5	71.3	2.48

Table 4: Results with GPT-2 Medium on E2E dataset. For all metrics other than the number of trainable parameters, higher values are better. Best results across PEFT methods are shown in **bold**, and second best results are underlined.

4.4 INSTRUCTION TUNING

Models and Datasets. Instruction tuning (Ouyang et al., 2022; Wei et al.; Mishra et al., 2022) involves fine-tuning models on paired prompts and responses. We apply VeRA, FourierFT and SiVA to LLaMA2 (Touvron et al., 2023), fine-tuning the LLaMA2-7B variant on the Alpaca dataset (Taori et al., 2023), which contains 51K instruction-following demonstrations. For evaluation, we generate responses to questions from MT-Bench (mtb), with GPT-4 scoring responses on a scale of 10. Since the GPT-4 model has likely changed since the baseline numbers were published, we evalute the generations from baseline methods and SiVA using the model behind the OpenAI API at the time of writing.

Implementation Details. For FourierFT, we use $n = 1000$ and use $r = 1024$ for VeRA. For our method, we use $perc = 0.997$. All methods are trained for one epoch. Hyperparameters are provided in Table 10 in the Appendix.

Model	Method	# Parameters	Score
LLAMA2 7B	VeRA	327K	**4.41**
	FourierFT	64K	4.31
	SiVA	**51K**	4.36

Table 5: Scores on MT-Bench Benchmark after tuning on the Alpaca dataset. Scores are provided by GPT-4 as judge.

Results. Table 5 shows the results for LLaMA2-7B. We only run on this variant due to compute and budget constraints. SiVA performs on par with other methods, with fewer parameters than either. Practical examples are presented in Appendix A.3.

4.5 ANALYSIS

Due to its inherent simplicity, our method comprises few components that can be altered or ablated. We concentrate on two primary aspects informed by our theoretical framework for this analysis: (1) The selection of a random set of singular vectors for constructing $\mathbf{\Delta W}$, rather than deriving them from $\mathbf{W}$ (referred to as "SiVA -Random"). This approach parallels the methodology of utilizing a random basis as seen in (Gao et al., 2024; Koohpayegani et al., 2024; Kopiczko et al., 2024); and (2) Training the upper singular values instead of the lower ones (designated as "SiVA -Top"). In both scenarios, we maintain the parameter count at each layer consistent with our standard formulation and use the same hyperparameters for training.

We report the results of this analysis in Table 6 for three image classification datasets using ViT-Base. Expectedly, we see a drop performance when we use random singular vectors, and when training the upper singular values.

Dataset	**EuroSAT**	**FGVC**	**OxfordPets**
SiVA	**98.25**$_{\pm 0.04}$	**47.43**$_{\pm 1.7}$	**95.16**$_{\pm 0.53}$
SiVA -Random	97.98	44.31	94.29
SiVA -Top	91.06	41.26	93.20

Table 6: Performance across diff datasets with random bases in ΔW and by training top singular values

5 CONCLUDING REMARKS

In this work, we introduce Singular Value Adaptation (SiVA), a novel, simple and efficient PEFT technique which is grounded in theoretical insights. SiVA increases the effective rank by selectively training only a subset of singular values in the adapter weight matrices, $\mathbf{\Delta W}$. The method's efficiency stems from two core principles: (1) *Sparsity*, where instead of updating all singular values of the pre-trained weight matrix $\mathbf{W}$, SiVA updates only the k-smallest singular values, with k determined adaptively at each layer based on the cumulative sum of singular values and a hyperparameter $perc$. This often leads to k being much smaller than the total number of singular values. (2) *Alignment*, where the singular vectors of the adapter matrix $\mathbf{\Delta W}$ are aligned with those of $\mathbf{W}$. This alignment reduces the need to train additional parameters, avoiding the necessity of training the singular vectors $U_{\mathbf{\Delta W}}$ and $V_{\mathbf{\Delta W}}$ as would be required with arbitrary updates. These principles allow SiVA to dramatically reduce the number of trainable parameters while maintaining or exceeding performance on large-scale datasets. Experimental results show SiVA achieves a $2\times$ to $50\times$ reduction in trainable parameters compared to existing PEFT methods like LoRA and FourierFT, while still achieving competitive performance. SiVA's low parameter count makes it especially advantageous for applications requiring frequent model switching or the storage of many fine-tuned models. Reuse of significant parts of the base model makes it ideal for tasks like domain adaptation, continual learning, and multi-task learning, where maintaining a backbone of shared information while adapting to specific downstream tasks is crucial.

As future work, SiVA's parameter efficiency on Transformer architectures can be linked to the architecture's inductive bias (Tarzanagh et al., 2023) and the cross-entropy loss. Additionally, exploring the relationship between SiVA's singular values and the intrinsic rank (Aghajanyan et al., 2021) could help clarify how closely SiVA approaches the optimal number of parameters required for each task.

Ethical Considerations and Reproducibility Statement

- **Human Subjects**: Our work does not make use of any human subjects. All datasets used in our work are publicly available and do not involve human subjects.
- **Results**: All results that we report are honestly executed and accurate to the best of our knowledge.
- **Potential Harmful Insights/Methods/Applications**: Our work does not have potential negative impact of harmful applications.
- **Research Integrity**: We have attempted to make best efforts to properly research existing works and ideas in the domain and have compared against contemporary benchmarks fairly.

We have provided the complete pseudocode of our approach in Algorithm 1 for purposes of reproducibility. Additionally, we provide all hyperparameters used for training our models in the Appendix (Tables 7, 9, 10, 8). We shall also release the complete code of our method used to obtain our reported results post acceptance.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7319–7328, 2021.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4133–4145, 2023.

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 *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Ali Edalati, Marzieh Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J Clark, and Mehdi Rezagholizadeh. Krona: Parameter efficient tuning with kronecker adapter.

Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. Parameter-efficient fine-tuning with discrete fourier transform. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=XUOHKSsurt.

SONG Haobo, Hao Zhao, Soumajit Majumder, and Tao Lin. Increasing model capacity for free: A simple strategy for parameter efficient fine-tuning. In *The Twelfth International Conference on Learning Representations*, 2024.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Xuehai He, Chunyuan Li, Pengchuan Zhang, Jianwei Yang, and Xin Eric Wang. Parameter-efficient model adaptation for vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 817–825, 2023.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

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, 2019a.

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, 2019b.

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 *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035, 2021.

Soroush Abbasi Koohpayegani, Navaneet K L, Parsa Nooralinejad, Soheil Kolouri, and Hamed Pirsiavash. NOLA: Compressing loRA using linear combination of random basis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=TjfXcDgvzk.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. VeRA: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NjNfLdxr3A.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

Tao Lei, Junwen Bai, Siddhartha Brahma, Joshua Ainslie, Kenton Lee, Yanqi Zhou, Nan Du, Vincent Zhao, Yuexin Wu, Bo Li, et al. Conditional adapters: Parameter-efficient transfer learning with fast inference. *Advances in Neural Information Processing Systems*, 36:8152–8172, 2023.

Brian Lester and Rami Al-Rfou Noah Constant. The power of scale for parameter-efficient prompt tuning.

Jonathan X Li, Will Aitken, Rohan Bhambhoria, and Xiaodan Zhu. Prefix propagation: Parameter-efficient tuning for long sequences. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

Baohao Liao, Yan Meng, and Christof Monz. Parameter-efficient fine-tuning without introducing new latency. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 1950–1965. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/0cde695b83bd186c1fd456302888454c-Paper-Conference.pdf.

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*, 2024.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks.

Ye Liu, Stefan Ultes, Wolfgang Minker, and Wolfgang Maier. Unified conversational models with system-initiated transitions between chit-chat and task-oriented dialogues. In *Proceedings of the 5th International Conference on Conversational User Interfaces*, pp. 1–9, 2023.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Ro{bert}a: A robustly optimized {bert} pretraining approach, 2020. URL https://openreview.net/forum?id=SyxS0T4tvS.

Fang Ma, Chen Zhang, Lei Ren, Jingang Wang, Qifan Wang, Wei Yu Wu, Xiaojun Quan, and Dawei Song. Xprompt: Exploring the extreme of prompt tuning. In *Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://api.semanticscholar.org/CorpusID:252780166.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3470–3487, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.244. URL https://aclanthology.org/2022.acl-long.244.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 487–503, 2021.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pp. 606–610, 2007.

G.W. Stewart. Perturbation theory for the singular value decomposition. UMIACS-TR-90-124, 1998/10/15/ 1998. URL http://drum.lib.umd.edu/handle/1903/552.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

Davoud Ataee Tarzanagh, Yingcong Li, Christos Thrampoulidis, and Samet Oymak. Transformers as support vector machines. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023. URL https://openreview.net/forum?id=gLwzzmh79K.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. Dylora: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3274–3287, 2023.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Tal Linzen, Grzegorz Chrupała, and Afra Alishahi (eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446.

Xiaolong Wang, Haipeng Yao, Tianle Mai, Song Guo, and Yun jie Liu. Reinforcement learning-based particle swarm optimization for end-to-end traffic scheduling in tsn-5g networks. *IEEE/ACM Transactions on Networking*, 31:3254–3268, 2023. URL https://api.semanticscholar.org/CorpusID:258941593.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Rui Hou, Yuxiao Dong, VG Vinod Vydiswaran, and Hao Ma. Idpg: An instance-dependent prompt generation method. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5507–5521, 2022.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2022.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*. Openreview, 2023.

Ruiyi Zhang, Rushi Qiang, Sai Ashish Somayajula, and Pengtao Xie. Autolora: Automatically tuning matrix ranks in low-rank adaptation based on meta learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5048–5060, 2024.

Zhen-Ru Zhang, Chuanqi Tan, Haiyang Xu, Chengyu Wang, Jun Huang, and Songfang Huang. Towards adaptive prefix tuning for parameter-efficient language model fine-tuning.

Wei Zhu and Ming Tan. Spt: Learning to selectively insert prompts for better prompt tuning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Yaoming Zhu, Jiangtao Feng, Chengqi Zhao, Mingxuan Wang, and Lei Li. Counter-interference adapter for multilingual machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2812–2823, 2021.