

RESEARCH ARTICLE

LoLDU: Low-Rank Adaptation via Lower-Diag-Upper Decomposition for Parameter-Efficient Fine-Tuning

Yiming Shi¹, Jiwei Wei¹, Yujia Wu¹, Ran Ran¹, Chengwei Sun¹¹ Center for Future Media and School of Computer Science and Engineering, University of Electronic Science and Technology of China, China

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.

Abstract

The rapid growth of model scale has necessitated substantial computational resources for fine-tuning. Existing approach such as Low-Rank Adaptation (LoRA) has sought to address the problem of handling the large updated parameters in full fine-tuning. However, LoRA utilize random initialization and optimization of low-rank matrices to approximate updated weights, which can result in suboptimal convergence and an accuracy gap compared to full fine-tuning. To address these issues, we propose LoLDU, a Parameter-Efficient Fine-Tuning (PEFT) approach that significantly reduces trainable parameters by 2600 times compared to regular PEFT methods while maintaining comparable performance. LoLDU leverages Lower-Diag-Upper Decomposition (LDU) to initialize low-rank matrices for faster convergence and orthogonality. We focus on optimizing the diagonal matrix for scaling transformations. To the best of our knowledge, LoLDU has the fewest parameters among all PEFT approaches. We conducted extensive experiments across 4 instruction-following datasets, 6 natural language understanding (NLU) datasets, 8 image classification datasets, and image generation datasets with multiple model types (LLaMA2, RoBERTa, ViT, and Stable Diffusion), providing a comprehensive and detailed analysis. Our open-source code can be accessed at <https://github.com/SKDDJ/LoLDU>.

Corresponding author: Jiwei Wei, mathematic6@gmail.com

I. Introduction

WITHIN the era of exponentially increasing the scale of models, fine-tuning these large models for new domains (e.g., Visual Instruction Tuning^[1]), applying advanced learning techniques (e.g., Representation Learning^{[2][3][4]}), or adapting to downstream tasks (e.g., Text-to-Image Customization^{[5][6]}, Object Tracking^{[7][8]}) requires substantial computational resources. To address this challenge, Parameter-Efficient Fine-Tuning (PEFT) techniques such as LoRA^[9], VeRA^[10], QLoRA^[11], and PiSSA^[12] have been developed to mitigate the bottleneck by reducing the number of trainable parameters, memory (VRAM), and storage costs.

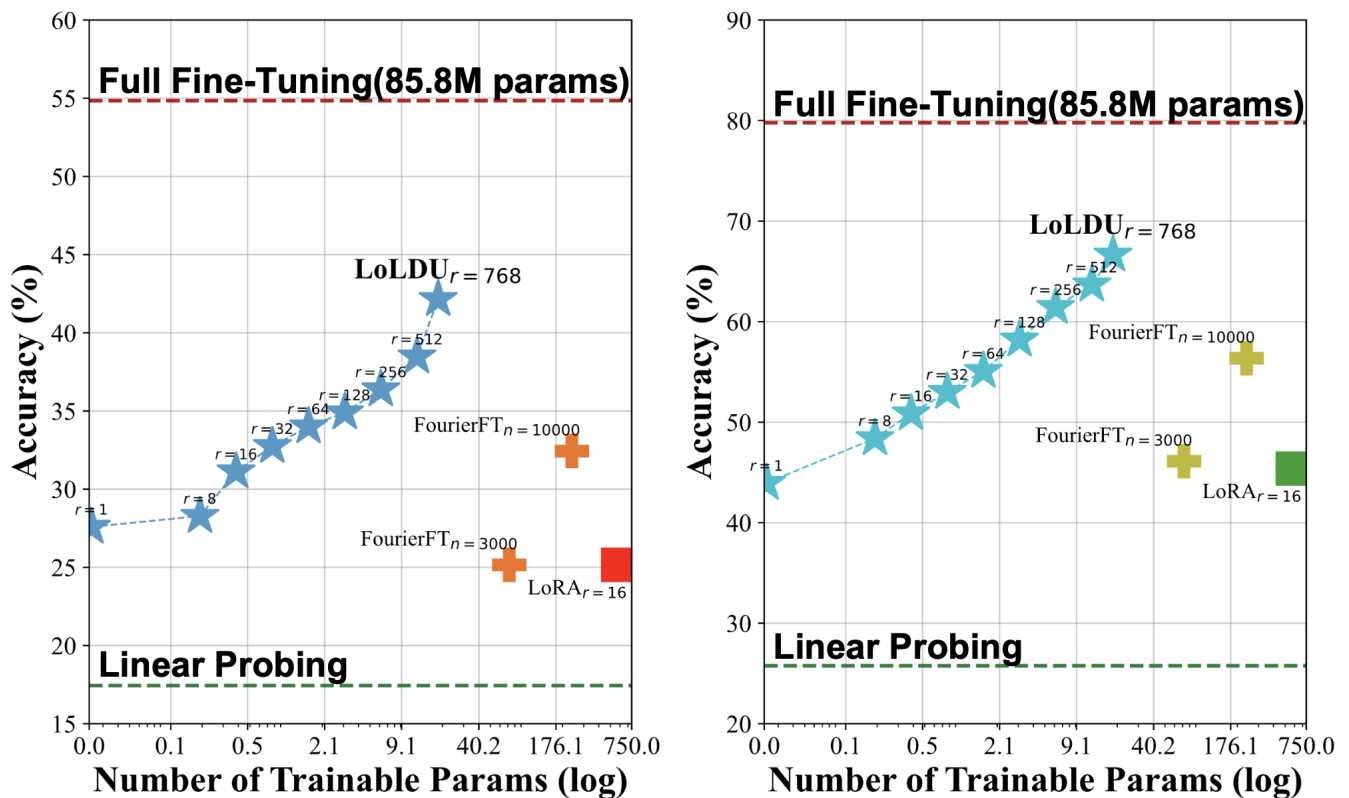


Figure 1. Performance vs log-scaled trainable parameters for FGVC (left) and StanfordCars (right) on ViT Base. Our LoLDU methods with $r = \{1, 8, 16, 32, 64, 128, 256, 512, 768\}$ exhibit superior parameter efficiency and performance when contrasted with Linear Probing [13] (LP, fine tuning the classifier head only¹), FourierFT [14] ($n = \{3000, 10000\}$), LoRA [9] ($r = 16$), and Full Fine-Tuning. LoLDU $r=768$ outperforms LoRA $r=16$ with 96.837% fewer trainable parameters. Particularly noteworthy is that LoLDU with $r = 1$ achieves competitive scores with just 24 trainable parameters, while LoLDU with $r = 768$ attains the highest accuracy: 42.15% for FGVC and 66.66% for StanfordCars, showcasing the scalability and effectiveness of our approach. Full Fine-Tuning (85.8M parameters) and Linear Probing represent the upper and lower performance bounds, respectively.

Despite advancements in PEFT, the process of fine-tuning large models remains prohibitively expensive in terms of both computational resources and storage requirements. For instance, fine-tuning a model with 7 billion parameters, such as LLaMA2 [15], on instruct-following tasks [16][17] incurs substantial costs. These costs are not limited to the training phase but extend to the storage of multiple fine-tuned model checkpoints, each consuming gigabytes of storage, thus leading to significant storage overhead. Approaches like Low-Rank Adaptation (LoRA) [9] and Vector-based Random Matrix Adaptation (VeRA) [10] have been developed to address these challenges by reducing the number of updated parameters. LoRA [9] achieves this by randomly initializing two low-rank matrices and optimizing them to approximate the model's updated weights. Similarly, VeRA [10] involves the random initialization and freezing of two matrices while training only two vectors for scale transformation. Recent research has revealed LoRA's limitations in data memorization due to low-rank updates. MoRA [18] addresses this issue through input dimension reshaping and square linear layer application. However, these methods often result in suboptimal convergence due to random initialization, as proposed by [19][20], thus yielding a provably small hyperspherical energy [21]. Furthermore, there is an accuracy gap compared to full fine-tuning, underscoring the need for more effective Parameter-Efficient Fine-Tuning strategies.

Thus, OFT^[21] proposes that maintaining orthogonality is crucial for preserving pre-trained knowledge, which enhances generalization^[22]. Building on this insight, we observe that Lower-Diag-Upper (LDU) decomposition inherently possesses orthogonal properties in its lower and upper triangular matrices. Additionally, we incorporate a heuristic initialization constrain the range of initialized values, resulting in a more stable training process.

In contrast to other PEFT approaches^{[9][12][10][18]}, which require fine-tuning $O(n^2)$ level parameters, for the first time, we demonstrate that it is possible to optimize only 0.00025% of parameters without any performance degradation. Our method, LoLDU, operates at $O(n)$ level and employs the LDU decomposition technique to extract the core model parameters, which are then fine-tuned for downstream tasks.

To demonstrate the efficiency of LoLDU across various model architectures, scales, and task types, we conduct an extensive set of experiments on tasks including instruction following^{[16][17][23]}, natural language understanding (NLU)^[24], image classification^{[25][26][27][28][29][30]}, and image generation^[6]. These experiments involved models with architectures such as LLaMA2-7B (decoder-only)^[15], RoBERTa-Base (encoder-decoder)^[31], ViT-Base (encoder-only)^[32], and Stable Diffusion^[33], with model scales ranging from 86 million to 7 billion parameters. This comprehensive evaluation verifies the effectiveness of our method across diverse scenarios.

In summary, this paper makes three key contributions:

- We introduce a novel approach to Parameter-Efficient Fine-Tuning (PEFT) by firstly attempting to leverage Lower-Diag-Upper (LDU) decomposition, offering a solution that maintains model performance while drastically reducing trainable parameters to as low as 0.00025% of the original model.
- We present LoLDU, a PEFT technique that harnesses Low-Rank Adaptation via Lower-Diag-Upper Decomposition, which operates with a complexity of $O(n)$. The LoLDU method employs orthogonal lower and upper triangular matrices to preserve pre-trained knowledge and enhance generalization, incorporating a heuristic initialization and scaling factor to optimize the diagonal matrix.
- LoLDU demonstrates the effectiveness and versatility through comprehensive experiments across various model architectures, scales, and task types. It offers a pioneering approach for efficient model adaptation across diverse scenarios in both NLP and CV domains.

II. Related Work

Parameter-Efficient Fine-Tuning (PEFT) is designed to mitigate the significant computational and storage costs associated with Full Fine-Tuning (FT). Among the various PEFT approaches, Low-Rank Adaptation (LoRA)^[9] offers a more flexible and generalized re-parameterization framework for fine-tuning, achieved by training two low-rank matrices to approximate the updated parameters. However, studies^{[19][20]} have indicated that random initialization for re-parameterization can be a bottleneck, leading to suboptimal convergence. In this work, we present the first attempt to address this issue by leveraging the Lower-Diag-Upper (LDU) decomposition technique for initialization. In Figure 2, we provide a comparison between LoRA and our LoLDU method.

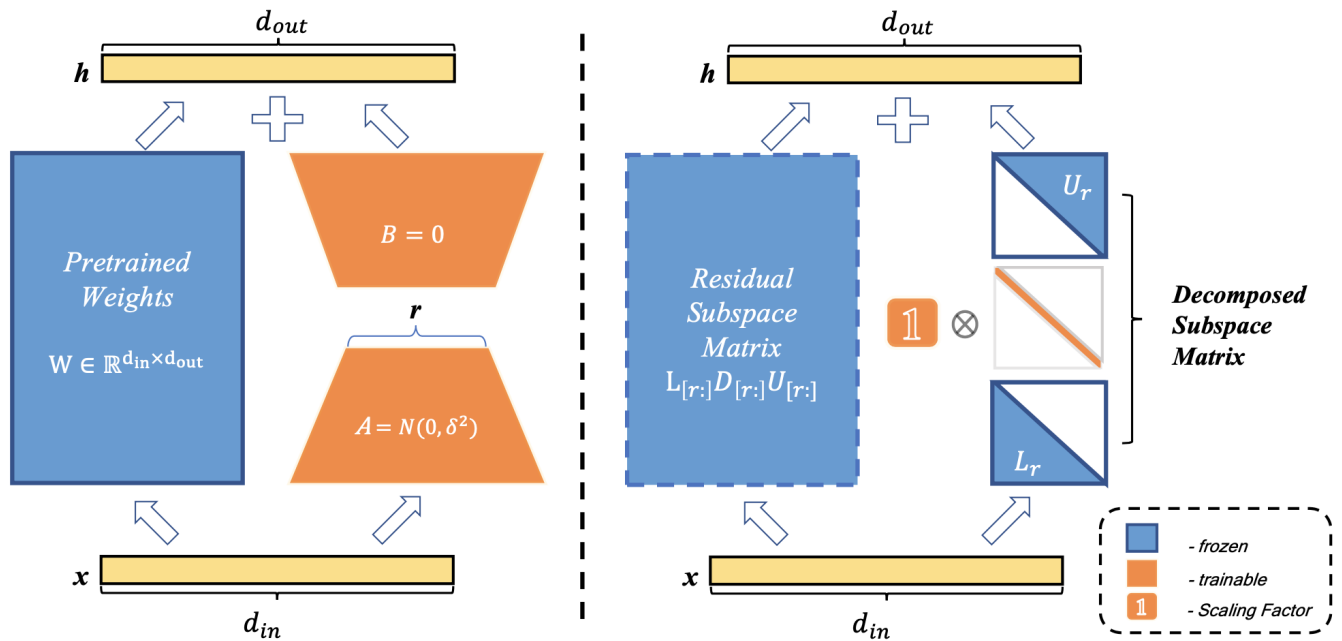


Figure 2. Comparison of LoRA (left) and our LoLDU (right) method. In LoRA, tunable parameters are low-rank (r) matrices A and B , with $\Delta W = BA$. For each weight W , there are $r \times (d_{in} + d_{out})$ trainable parameters. LoLDU, however, optimizes a diagonal matrix for scale transformation, preserving original model knowledge during tuning. The weight update in LoLDU is $\Delta W = \sigma \cdot P \cdot (L_r \text{diag}(z_r), U_r)$, involving $r + 1$ trainable parameters. The permutation matrix P , while omitted in this figure for simplicity, is included in Figure 3.

Parameter efficient fine tuning (PEFT). To date, existing PEFT approaches can be divided into three categories: (1)

Additive PEFT: This approach introduces new tunable parameters or modifies model representations. Examples include adapters^{[34][35][36][37]} and prefix-tuning^[38], which add small, trainable components to the model for efficient task-specific learning. (2) **Selective PEFT**^{[39][40][41][42]}: This method fine-tunes only a subset of the model's parameters, such as specific layers or neurons. Techniques like BitFit^[43] aims to only update bias parameters b , while maintaining fixed weights W , to shift the model's conditional distribution $p(y|x; \theta)$ towards the target domain distribution $p_{\text{target}}(y|x)$, where θ denotes the model parameters. (3) **Re-parameterized PEFT**^{[9][44][45]}. This technique usually reconstructs model parameters in a low-dimensional space as new knowledge is often represented in a low-rank form^[46].

Low-Rank Adaptation. LoRA^[9] decomposes parameter matrices into low-rank forms, maintaining performance while reducing the number of parameters to be fine-tuned. Previous studies have credited LoRA for its efficiency in inference and storage, albeit at an expensive training cost due to the random initialization, which causes the model to saturate more slowly. Recent studies^[47] have attempted to bridge this gap by exploring the development of new initialization methods to create LoRA parameters instead of starting from scratch. Advancing the initialization strategies for LoRA parameters is imperative for enhancing the quality and adaptability of downstream tasks. Therefore, Section IV delves into the exploration of various initialization methodologies.

Re-parameterization. Singular Value Decomposition (SVD) is widely utilized for re-parameterization in Parameter-Efficient Fine-Tuning (PEFT) methods. Recent studies^{[12][48][49][37][50]} have explored various SVD-based approaches for

low-rank matrix initialization. These include fine-tuning singular values of reshaped weight matrices^[50], initializing adapter matrices with principal components^[12], introducing intermediate matrices between frozen principal components matrices, and updating weights as sparse combinations of singular vector outer products^[49]. However, SVD's computational complexity $O(mn^2 + n^3)$ for an $m \times n$ matrix remains a constraint compared to LDU decomposition $O(mn^2 - n^3/3)$. Furthermore, LDU decomposition offers a more interpretable representation of matrix structure through elementary row operations and pivoting strategies.

III. Method

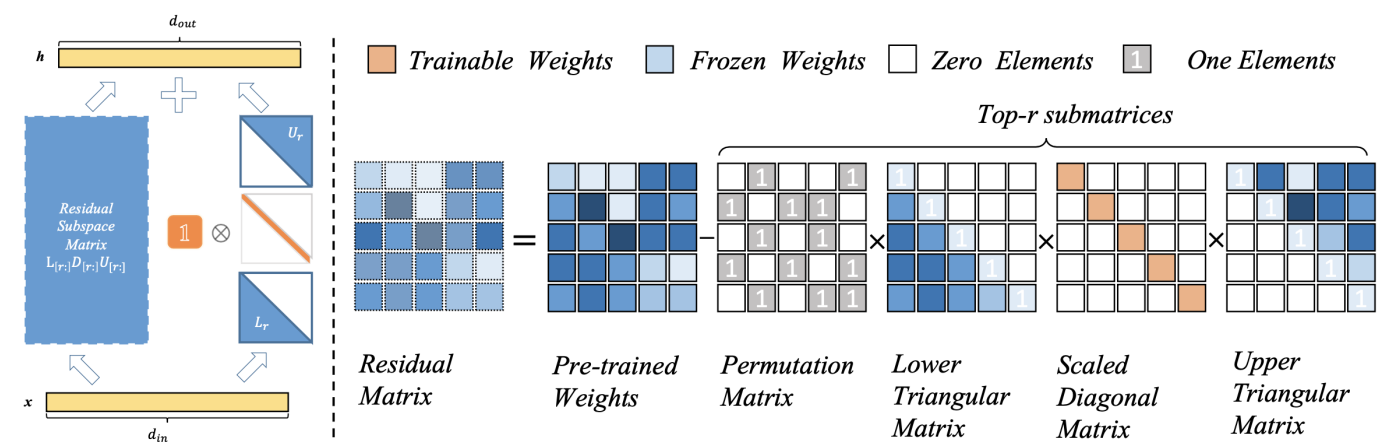


Figure 3. Schematic representation of our LoLDU method. The left diagram illustrates the forward pass, demonstrating the transformation of the input $x \in \mathbb{R}^{d_{in}}$ into the output $h \in \mathbb{R}^{d_{out}}$ via a residual subspace matrix $L_{[r]} D_{[r]} U_{[r]}$ and a decomposed subspace matrix $\sigma L_r D_r U_r$. The right diagram shows the initialization process, where the residual matrix is obtained by performing LDU decomposition on the pre-trained weights, then subtracting the top- r submatrices (top- r rows and columns) from the permutation matrix (P), lower triangular (L), scaled diagonal (D), and upper triangular (U) matrices. Diagonal matrix is trainable (orange), while the other matrices remain fixed (blue). LoLDU enables efficient adaptation of pre-trained models via low-rank updates, reducing both computational cost and parameter count.

We present LoLDU (depicted in Figure 3), a parameter-efficient-fine-tuning method utilizing Lower-Diag-Upper (LDU) decomposition. LoLDU builds upon the principle proposed by LoRA^[9], focusing on learning the changes in pre-trained weights. In contrast to LoRA, which employs random initialization, LoLDU leverages the LDU decomposition for initialization. We then compute the Residual Subspace Matrix (RSM) by applying element-wise subtraction of the Decomposition Subspace Matrix (DSM) from the original matrix. The DSM is constructed using the first r entries, which are selected to maintain a low-rank formation while remaining trainable.

A. Initialization and Orthogonal Space Preservation

Previous works have shown that maintain the orthogonality nature is crucial to improve the representation quantity^[21]. The advantage of LDU decomposition is the factorization that preserves the orthogonality of the lower and upper triangular matrices. We leverage this property to initialize the low-rank matrices. The LDU decomposition factorizes a

matrix $W_0 \in \mathbb{R}^{m \times n}$ into four matrices:

$$W_0 = P \cdot L \cdot \text{diag}(z) \cdot U,$$

where $P \in \mathbb{R}^{m \times m}$ is a permutation matrix, $L \in \mathbb{R}^{m \times k}$ is lower triangular with ones on the diagonal, $\text{diag}(z) \in \mathbb{R}^{k \times k}$ is the diagonal formation of vector z , and $U \in \mathbb{R}^{k \times n}$ is upper triangular with ones on the diagonal, where $k = \min(m, n)$. This property is essential for obtaining an equivalent formation to the original model weight W_0 .

Specifically, we optimize only the diagonal entries of $\text{diag}(z)$ and dynamically adjust the scaling factor σ to align updated parameters with the target matrix, wherein the σ is initialized to 1.0.

B. Low-Rank Approximation

In the realm of learning weight changes, our approach aligns with the principles of LoRA-based methods^{[9][18][51][52]}, which mitigate inference latency by merging pre-trained weights with the learned adapter matrices.

Formally, let $W_0 \in \mathbb{R}^{m \times n}$ represent the pre-trained weight matrix, and $\Delta W \in \mathbb{R}^{m \times n}$ denote the weight changes introduced during fine-tuning. LoRA parameterizes ΔW using a low-rank decomposition in the forward pass:

$$h = W_0 x + \Delta W x = W_0 x + B A x,$$

where $B \in \mathbb{R}^{m \times r}$ and $A \in \mathbb{R}^{r \times n}$ are trainable matrices, with the rank $r \ll \min(m, n)$.

In contrast, our proposed method, LoLDU, decomposes the weight matrix W_0 using an LDU (Lower-Diag-Upper) decomposition which breaks down W_0 into four matrices: $\sigma \cdot P \cdot (L, \text{diag}(z), U)$. We take the inspiration from^{[53][46]} that learned adapter matrices reside in a low intrinsic dimension. Therefore, we extract the top r components from the LDU decomposition, which helps in maintaining an intrinsic subspace to adapt to downstream tasks. These components are represented as follows:

$$B = L_r = L_{[:, :r]} \in \mathbb{R}^{m \times r}, \quad (3)$$

$$\text{diag}(z_r) = D_{:, r, :r} \in \mathbb{R}^{r \times r}, \quad (4)$$

$$A = U_r = U_{:, r, :} \in \mathbb{R}^{r \times n}, \quad (5)$$

where L_r represents the first r columns of the lower triangular matrix L , $D_{:, r, :r}$ denotes the top r by r block of the diagonal matrix D , and U_r is the first r rows of the upper triangular matrix U . These components capture the essential structure of the original weight matrix in a reduced form.

C. LoLDU Weight Adaptation Procedure

Using these components, we define the Decomposed Subspace Matrix (DSM), which reconstructs a part of the original weight matrix using the top r components. The DSM is formulated as:

$$\text{DSM} = \sigma \cdot P \cdot (L_r \text{diag}(z_r), U_r), \quad (6)$$

where σ is introduced to control the magnitude of the weight updates as a scaling factor.

Next, we obtain the Residual Subspace Matrix (RSM) by subtracting the DSM from the original weight matrix W_0 , which ensures that the RSM captures the information not represented by the top r components, thereby preserving the full knowledge encoded in W_0 :

$$\text{RSM} = W_0 - \text{DSM}. \quad (7)$$

The weight change ΔW is parameterized as:

$$\Delta W = \text{DSM} = \sigma \cdot P \cdot (L_r \text{diag}(z_r), U_r), \quad (8)$$

by parameterizing ΔW in this manner, efficient updates to the model weights are enabled without significantly increasing the parameter count.

The advantage of LoLDU lies in its use of orthogonal, lower, and upper triangular matrices, which help preserve the inherent knowledge of the model. The orthogonal nature of these matrices ensures that the decomposed components maintain their properties during transformations as proposed by^[22], thereby preserving the information integrity. Moreover, we initialize $\text{diag}(z_r)$ using heuristic methods such as Constant (D_r mean), Uniform, Normal, or Regular LDU, to enhance training stability.

The proposed forward pass can be expressed as follows:

$$\begin{aligned} h &= \text{RSM}x + \Delta Wx \\ &= \text{RSM}x + \text{DSM}x \\ &= \text{RSM}x + \sigma \cdot P \cdot (L_r \text{diag}(z_r), U_r)x. \end{aligned} \quad (9)$$

Algorithm 1 Low-Rank LDU Decomposition and Optimization for Layer Weight Adaptation

Input: Weight matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$, rank r , alpha α , learning rate η , number of iterations T , projection operator \mathcal{P}

Output: Decomposed components \mathbf{P} , \mathbf{L}_r , \mathbf{D}_r , \mathbf{U}_r , residual $\mathbf{W}_{\text{residual}}$, scaling factor σ , optimized \mathbf{D}_r , optimized σ

- 1: **Phase 1: Initial Decomposition**
 - 2: $\mathbf{P}, \mathbf{L}, \mathbf{U} \leftarrow \text{LU_decomposition}(\mathbf{W})$ // Perform standard LU decomposition
 - 3: $\mathbf{D} \leftarrow \text{diag}(\mathbf{U})$ // Extract diagonal matrix
 - 4: $\mathbf{U} \leftarrow \mathbf{D}^{-1} \mathbf{U}$ // Normalize U
 - 5: **Phase 2: Low-Rank Approximation**
 - 6: $\mathbf{L}_r \leftarrow \mathbf{L}_{:,1:r}$ // Extract first r columns of L
 - 7: $\mathbf{D}_r \leftarrow \mathbf{D}_{1:r,1:r}$ // Extract top-left $r \times r$ submatrix of D
 - 8: $\mathbf{U}_r \leftarrow \mathbf{U}_{1:r,:}$ // Extract first r rows of U
 - 9: **Phase 3: Scaling Factor and Residual Computation**
 - 10: $\sigma \leftarrow \alpha/r$ // Compute scaling factor
 - 11: $\mathbf{W}_{\text{approx}} \leftarrow \sigma \mathbf{P} \mathbf{L}_r \mathbf{D}_r \mathbf{U}_r$ // Compute low-rank approximation
 - 12: $\mathbf{W}_{\text{residual}} \leftarrow \mathbf{W} - \mathbf{W}_{\text{approx}}$ // Compute residual matrix
 - 13: **Phase 4: Heuristic Initialization**
 - 14: Apply heuristic initialization to \mathbf{D}_r // Choose from methods: Constant($(\mathbf{D}_r.\text{mean})$), Uniform, Normal, or Regular LDU
 - 15: **Phase 5: Optimization with Projected Gradient Descent**
 - 16: **for** $t \leftarrow 1$ to T **do**
 - 17: Compute gradients $\nabla_{\mathbf{D}_r} \mathcal{L}$ and $\nabla_{\sigma} \mathcal{L}$
 - 18: $\mathbf{D}_r \leftarrow \mathcal{P}(\mathbf{D}_r - \eta \cdot \nabla_{\mathbf{D}_r} \mathcal{L})$
 - 19: $\sigma \leftarrow \mathcal{P}(\sigma - \eta \cdot \nabla_{\sigma} \mathcal{L})$
 - 20: **end for**
 - 21: **return** $\mathbf{P}, \mathbf{L}_r, \mathbf{D}_r, \mathbf{U}_r, \mathbf{W}_{\text{residual}}, \sigma$
-

Algorithm 1.

D. Optimization Process

The fine-tuning phase of LoLDU employs a sophisticated optimization strategy, focusing on the diagonal matrix \mathbf{D}_r and the scaling factor σ . This approach represents a departure from conventional fine-tuning methods, offering more granular control over parameter updates while preserving the integrity of pre-trained knowledge.

The optimization problem is formulated as a constrained minimization:

$$\begin{aligned}
& \underset{D_r, \sigma}{\text{minimize}} && L(f_{W_0 + \Delta W}(x), y) \\
& \text{subject to} && \|D_r\|_F \leq \epsilon, \quad (10) \\
& && 0 < \sigma \leq 1,
\end{aligned}$$

where L denotes the task-specific loss function, $f_{W_0 + \Delta W}$ denotes the model with updated weights, (x, y) are the input-output pairs from the fine-tuning dataset, $\|\cdot\|_F$ represents the Frobenius norm, and ϵ is a set constraint threshold.

To address the constrained nature of the optimization problem, we employ a projected gradient descent method, ensuring that updates to D_r and σ remain within the feasible region defined by the constraints. This is achieved through a projection operator P :

$$\begin{aligned}
D_r^{t+1} &= P \left(D_r^t - \eta_t \frac{\partial L}{\partial D_r^t} \right), \\
\sigma^{t+1} &= P \left(\sigma^t - \eta_t \frac{\partial L}{\partial \sigma^t} \right),
\end{aligned}$$

where η_t is the learning rate at iteration t , adaptively adjusted using techniques such as Adam^[54] or RMSprop^[55] to account for the geometry of the parameter space.

Please refer to Algorithm 1 for additional detailed information.

E. Computational Complexity Analysis

The computational efficiency of LoLDU can be evaluated in terms of both space and time complexity:

Space complexity: The storage requirement for LoLDU is $O(r + 1)$, which is considerably lower than the $O(mr + m)$ required by methods such as LoRA. This reduction in parameter count not only leads to significant memory savings but improves efficiency during both the training and inference phases.

Time complexity: The forward pass of LoLDU requires $O(mnr)$ operations with a minor linear term $O(r)$. In contrast to methodologies that necessitate recurrent complex iterations^{[56][52]}, LoLDU performs the LDU decomposition only once during initialization, with a time complexity of $O(mn^2 - n^3/3)$, and utilizing direct updates via projected gradient descent without iterative refinement, ensuring efficient parameter optimization and rapid convergence.

In summary, LoLDU leverages LDU decomposition to efficiently parameterize weight changes, reducing the number of tunable parameters and maintaining high performance. This method provides a more efficiency and effective alternative to traditional LoRA-based approaches.

IV. Experiments

Model	Method	# Params	SST-2 acc	MRPC acc	CoLA cor	QNLI acc	RTE acc	STS-B cor	Avg.
RoBERTa-Base	FT	125M	94.8	90.2	63.6	92.8	78.7	91.2	85.2
	BitFit	0.1M	93.7	92.7	62.0	91.8	81.5	90.8	85.4
	LoRA	0.3M	95.1	89.7	63.4	93.3	78.4	91.5	85.2
	PiSSA	0.707M	94.6	88.4	63.0	93.1	85.9	91.2	86.0
	VeRA	0.043M	94.6	89.5	65.6	91.8	78.7	90.7	85.2
	LoLDU	0.0184M	94.8	89.9	63.8	92.9	81.3	92.3	85.8
	$\Delta_{baseline}$	6.13%	-0.3	+0.2	+0.4	-0.4	+2.9	+0.8	+0.6

Table I. Results for different adaptation methods on the GLUE benchmark. The term "Params" refers to the number of parameters updated during fine-tuning. We report Matthew's correlation for CoLA, Pearson correlation for STS-B, and accuracy for the remaining tasks. Higher values indicate better performance. Except LoLDU, all results are from prior work. LoLDU performs on par with LoRA while using significantly fewer parameters. The $\Delta_{baseline}$ row shows the percentage increase or decrease in performance compared to our method.

This section presents an evaluation of LoLDU within the fields of natural language processing (NLP) and computer vision (CV). For NLP, LoLDU is applied for fine-tuning: (1) RoBERTa Base^[31] on natural language understanding (GLUE^[24]), and (2) LLaMa-2 7B^[15] on instruction tuning (Alpaca^[16], Vicuna^[17]). For CV, we apply LoLDU to fine-tune: (1) Vision Transformers (ViT) Base^[32] on image classification^{[25][26][27][28][29][30]}, and (2) Stable Diffusion v1.5^[33] on customized image generation^[6].

We compare our LoLDU method with widely used Parameter-Efficient Fine-Tuning (PEFT) methods. To ensure a fair comparison, we replicate the setups from previous studies^{[9][14][57]} and utilize their reported results.

The baselines considered are:

- **Full Fine-Tuning (FT):** FT trains all model parameters on the task-specific data.
- **LoRA^[9]:** LoRA updates weights by injecting two tunable low-rank matrices for parameterization.
- **MELoRA^[57]:** MELoRA trains a group of mini LoRAs to maintain a higher rank.
- **FourierFT^[14]:** FourierFT learns a small fraction of spectral coefficients using the Fourier transform.

Finally, we perform ablation studies to examine the impact of initialization methods, scaling factors, and rank. Further results concerning the learning rate and rank are detailed in Appendix E1 and Appendix E2. We conduct all experiments on a single NVIDIA RTX A6000 (48G) GPU.

Method	Mean Acc.	Params (%)	Keep Orthogonal	No random Init.	No extra Infer. cost	Faster convergence
FullFT	88.20	100	✗	✓	✓	✓
LP	68.38	-	✗	✗	✓	✗
LoRA	76.22	6.77	✗	✗	✓	✗
FourierFT	79.29	2.79	✗	✗	✓	✗
LoLDU	82.79	0.21	✓	✓	✓	✓

Table II. Comparative analysis of various methods on image classification datasets using ViT Base models. The table reports the mean accuracy (%) after 10 epochs, alongside parameters efficiency and approach features.

A. Natural Language Understanding

a) Models and Datasets

We evaluate LoLDU on the GLUE benchmark (General Language Understanding Evaluation^[24]), which comprises nine NLU tasks. These tasks include single-sentence classification (CoLA, SST-2), similarity and paraphrasing (MRPC, STS-B, QQP), and natural language inference (MNLI, QNLI, RTE, WNLI). For evaluation, we fine-tune pre-trained RoBERTa Base models^[31].

b) Implementation Details

We adopt the experimental setup of VeRA^[10], tuning the hyperparameters for learning rates and the scaling factor values across six datasets in the GLUE benchmark. Following the approach of LoRA^[9], we fully fine-tune the classification head. We apply LoLDU to the weight matrices W_q , W_k , W_v , and W_o in each transformer block. Hyperparameters are provided in Table VII in the Appendix.

c) Results

Results are summarized in Table I. Following^{[9][52]}, and^[58], we specify the number of trainable parameters for the fine-tuned layers excluding the classification head. We report the median of five random seed results, selecting the best epoch for each run. In general, LoLDU achieves better or on-par performance compared to baseline methods with significantly fewer trainable parameters. Notably, LoLDU outperforms all baselines including fully fine-tuning the RoBERTa Base on STS-B. As mentioned in Section III, the parameter count of LoRA is dependent on both the width and depth of models, resulting in a larger count growth (LoRA: 0.3M; ours: 0.0184M) compared to LoLDU.

B. Instruction Tuning

a) *Models and Datasets*

Instruction tuning^{[59][60][16]} is a technique that involves fine-tuning large language models (LLMs) on paired data consisting of instructions and their corresponding outputs to enhance the quality of the model's responses. In our study, we apply LoRA^[9] and LoLDU to fine-tune the LLaMA2 model^[15]. Specifically, we use LLaMA2-7B as the base model, which is then fine-tuned on the Alpaca dataset^[16]. This dataset comprises 52,000 instruction-output pairs generated by OpenAI's text-davinci-003 model. For evaluation, we conduct a rigorous and holistic assessment of the fine-tuned model using INSTRUCTEVAL^[23], allowing us to systematically analyze the model's performance in problem-solving, writing ability, and alignment to human values.

b) *Implementation Details*

In the implementation of LoRA, a rank of $r = 64$ is employed, with a focus on updating all linear layers, excluding the language modeling head (lm_head), and specifically targeting the W_Q and W_V matrices. For LoLDU, the training process spans three epochs, and we present the average performance scores across all evaluated responses. Hyperparameter configuration is detailed in Table VIII in Appendix B.

c) *Results*

The results, as presented in Table III, demonstrate that LoLDU achieves a slight improvement over the performance of LoRA, while employing merely 0.05% of the parameters required by LoRA.

Model	Method	# Params	MMLU	DROP	HEval	BBH
LLaMA2-7B	w/o FT	-	45.96	31.55	12.20	32.04
	LoRA	33.6M	45.64	32.46	15.09	32.40
	AdaLoRA	33.6M	45.96	31.94	14.02	32.85
	MELoRA	0.5M	46.46	32.65	16.16	33.01
	LoLDU	0.016M	46.21	32.71	15.11	33.12
	$\Delta_{baseline}$	0.05%	+0.57	+0.25	+0.02	+0.72

Table III. Results on INSTRUCTEVAL for instruction-following tasks: exact match for MMLU, DROP, and BBH, pass@1 for HumanEval. Higher values are preferable. Boldface indicates the best metric values. The $\Delta_{baseline}$ row displays the performance change percentage compared to our method.

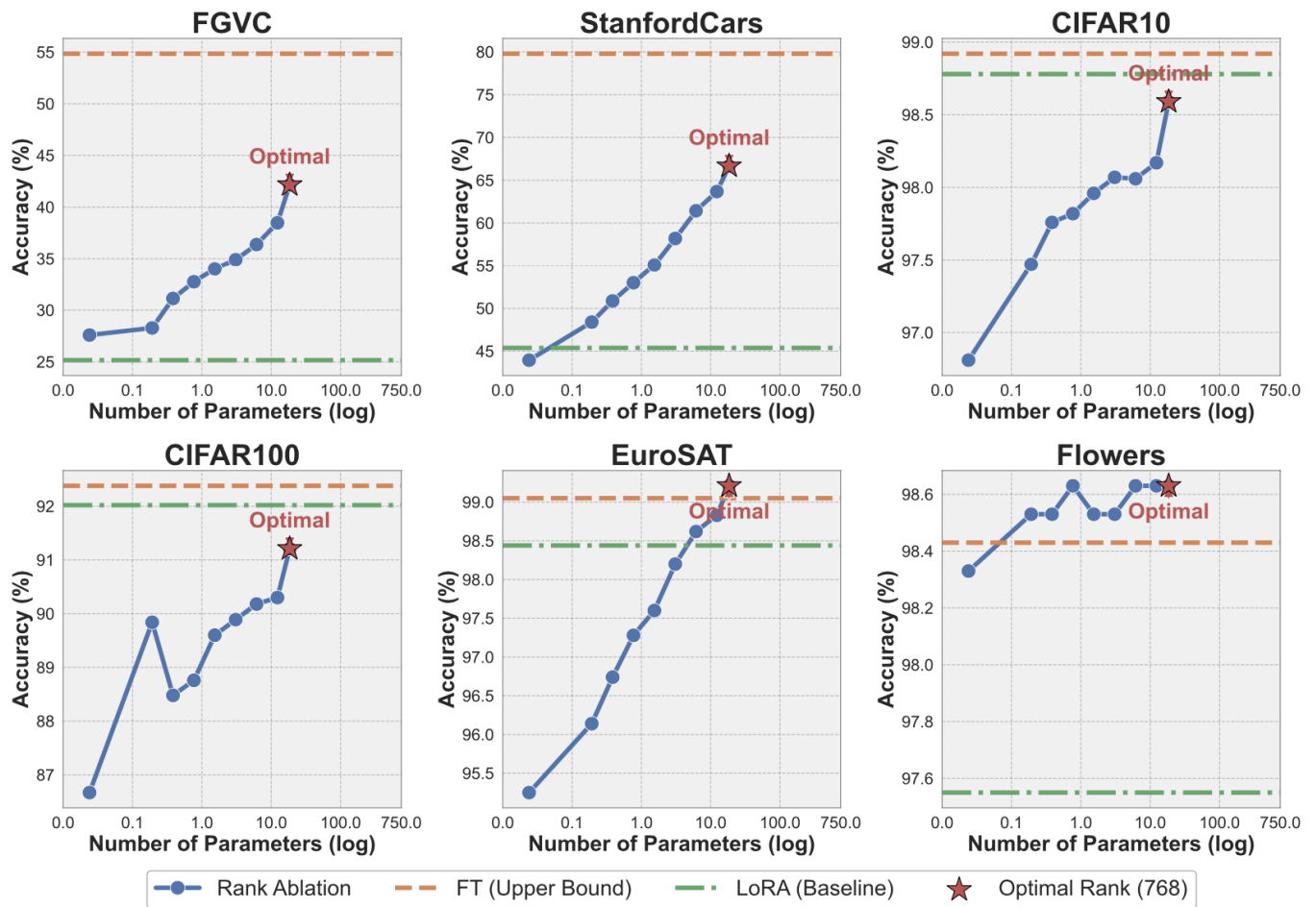


Figure 4. Comprehensive Analysis of Rank Ablation Study Results. This figure presents the performance of the ViT-base model on various image classification tasks using the LoLDU method with different ranks. The x-axis shows ranks (1 to 768), and the y-axis indicates accuracy for datasets: FGVC, StanfordCars, CIFAR10, CIFAR100, EuroSAT, and Flowers.

C. Image Classification

a) Models and Datasets

We assess our approach on image classification utilizing the Base version of the Vision Transformer (ViT)^[32], pre-trained on ImageNet-21K^[61]. Fine-tuning is performed on datasets such as CIFAR10 (10)^[25], EuroSAT (10)^[30], as well as StanfordCars (196)^[28], FLOWERS102 (102)^[27], FGVC (100)^[29], and CIFAR100 (100)^[26], covering both small and large label spaces. For detailed information, refer to Appendix C.

b) Implementation Details

We include three baselines for evaluation: Full Fine-Tuning (FT), Linear Probing^[13] (LP, fine-tuning the classification head only), and LoRA^[9]. We adhere to the experimental configurations established by FourierFT^[14]. For both LoRA and our method, only the W_Q and W_V matrices of ViT are updated. We use $r = 16$ for LoRA and $r = \{64, 768\}$ for LoLDU. Detailed hyperparameter configurations are available in Table IX in the Appendix C.

Model	Method	# Params	FGVC acc	StanfordCars acc	CIFAR10 acc	CIFAR100 acc	EuroSAT acc	Flowers acc	Avg.
ViT-Base	LP	-	17.44	25.76	96.41	84.28	88.72	97.64	68.38
	FT	85.8M	54.84	79.78	98.92	92.38	99.05	98.43	87.23
	LoRA(r16)	581K	25.16	45.38	98.78	92.02	98.44	97.55	76.22
	FourierFT(‡)	72K	27.51	46.11	98.58	91.20	98.29	98.14	76.64
	FourierFT(†)	239K	32.44	56.36	98.69	91.45	98.78	98.04	79.29
	LoLDU(r64)	1.5k	32.31	50.99	97.96	89.60	97.60	98.53	77.83
	LoLDU(r768)	18k	42.15	66.66	98.59	91.21	99.21	98.92	82.79
	$\Delta_{baseline}$	3.173%	+16.99	+21.28	-0.19	-0.81	+0.77	+1.37	+6.57

Table IV. We conducted a comparison on image classification datasets using ViT Base models. The accuracy (%) after 10 epochs is reported.

FourierFT was evaluated using different trainable parameters for each layer, indicated by symbols: (‡) for 3000 and (†) for 10000. $\Delta_{baseline}$ represents the performance gap between our LoLDU method and the baseline method LoRA. Bold denotes the best results.

c) Results

Table IV presents the results for six image classification datasets using the ViT Base model. LoRA and LoLDU demonstrate superior performance compared to Linear Probing^[13], showcasing their efficacy in image classification tasks within the computer vision domain. Notably, our approach achieves comparable outcomes while utilizing merely 3.173% of LoRA's parameters. LoLDU exhibits particularly impressive gains, surpassing LoRA by 15.28% and 16.99% in FGVC and StanfordCars tasks, respectively, effectively narrowing the accuracy gap with Full Fine-Tuning, as depicted in Figure 1. Furthermore, LoLDU outperforms all baselines, including Fully Fine-Tuning, on EuroSAT and Flowers datasets.

D. Image Generation

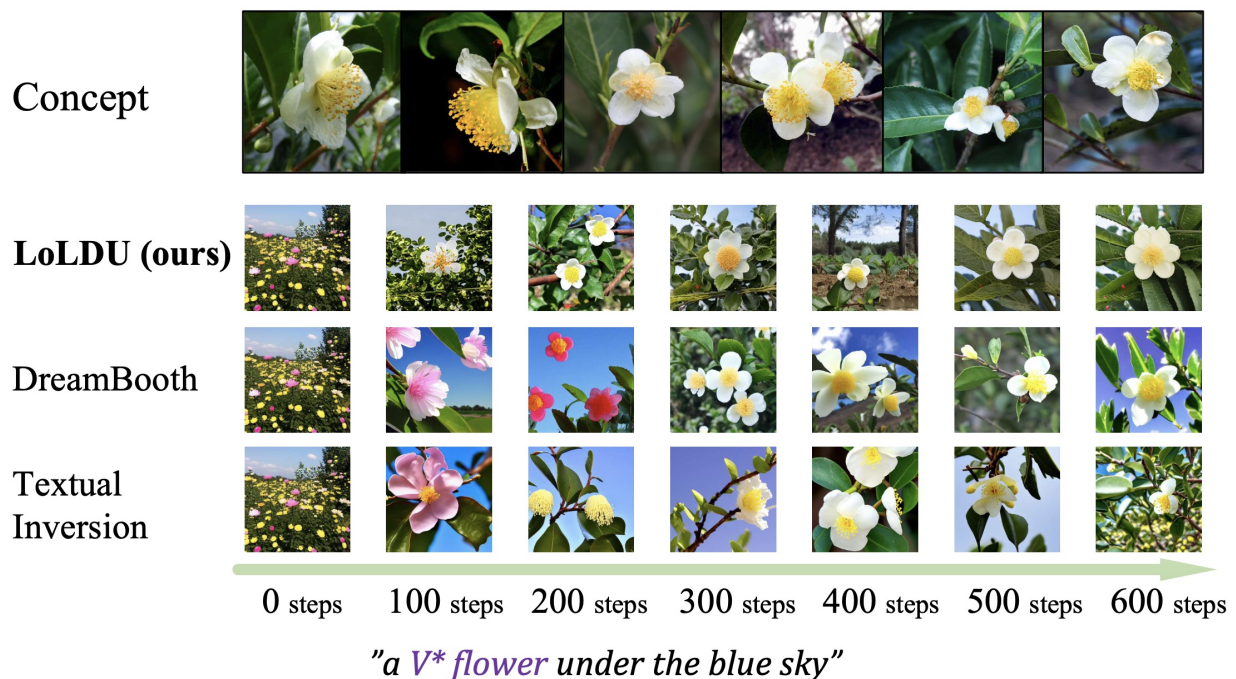


Figure 5. Concept Learning Progression In Text-to-Image Generation. Top row: target concept. Subsequent rows: generated images using LoLDU (our method), DreamBooth^[6], and Textual Inversion^[5], respectively, at training steps 0-600. LoLDU exhibits accelerated convergence, achieving concept acquisition within ~ 100 steps, surpassing baseline methods in efficiency.

a) Models and Datasets

We assess our method in the domain of image generation. Recent research^{[5][6]} highlights the necessity for customization in this field, which holds significant practical implications. The goal is to fine-tune a text-to-image model using a limited set (typically 3-5) of images representing an unique concept (e.g., a scene, individual, pet, or object) to effectively capture and reproduce the novel concept. For this study, we employ the v1.5 version of Stable Diffusion (SD)^[33], a widely-adopted computer vision foundation model. SD is pre-trained on LAION-5B^[62], a dataset consists of 5.85 billion image-text pairs filtered using CLIP^[63].

b) Implementation Details

We conduct our experiments on seven different concepts, including persons, pets, and objects, using the CustomConcept101 dataset^[64] and the human-centric FFHQ dataset^[65]. We select two concurrent works as baselines: Textual Inversion^[5] and DreamBooth^[6]. Textual Inversion learns new concept by mapping it from the image to the textual modality, encoding them as a rare token in the embedding space. DreamBooth, utilizes a semantic prior (e.g., class-specific) to maintain the subject's key features. We provide the datasets in Figure 6 and hyperparameters in Table X in Appendix D.

Initialization Method	FGVC acc	StanfordCars acc	CIFAR10 acc	CIFAR100 acc	EuroSAT acc	Flowers acc	Avg.
ViT-Base Initialization Ablation Study							
Uniform(†)	2.37 / 2.37	1.17 / 1.38	35.92 / 28.93	14.22 / 9.71	57.81 / 52.95	4.51 / 4.41	19.33 / 16.63
Normal(✚)	39.60 / 39.12	65.17 / 65.00	98.02 / 98.33	90.27 / 90.54	99.00 / 99.03	98.63 / 98.63	81.78 / 81.78
Normal(★)	2.10 / 2.13	1.34 / 1.12	29.17 / 26.54	10.11 / 7.91	52.98 / 48.49	4.61 / 4.41	16.72 / 15.10
Constant(z.mean)	42.21 / 41.16	65.41 / 63.86	98.38 / 98.21	90.77 / 90.21	99.16 / 98.99	98.63 / 98.43	82.43 / 81.81
Zeros	9.30 / 9.24	8.27 / 9.09	72.43 / 72.13	46.00 / 43.27	96.44 / 96.05	41.08 / 40.49	45.59 / 45.05
Ones	2.01 / 1.95	1.16 / 1.16	30.89 / 26.26	10.29 / 8.60	50.95 / 46.61	3.73 / 4.41	16.51 / 14.83
Regular LDU	40.50 / 40.44	65.12 / 62.37	98.28 / 98.20	90.61 / 90.61	99.04 / 98.95	98.92 / 98.92	82.08 / 81.58
Uniform(‡)	42.15 / 39.72	66.66 / 64.54	98.59 / 98.28	91.21 / 90.48	99.21 / 98.97	98.63 / 98.82	82.74 / 81.80

Table V. Ablation study of different initialization methods across six image classification datasets. We set rank up to 768 and learning rate to $3e-3$ and test on the ViT base model. The datasets include FGVC, StanfordCars, CIFAR10, CIFAR100, EuroSAT, and Flowers. The uniform initialization method is indicated by symbols: ‡ for (a=-1, b=1) and † for (a=-z.mean/2, b=z.mean/2). The normal initialization method is indicated by symbols: ✚ for (mean=0, std=1) and ★ for (mean=z.mean, std=z.std). For each entry, the left value represents results with scaling factor, while the right value in gray represents results without scaling factor. The average performance (Avg.) across all datasets is also reported. Bold denotes the best results for each dataset and the average.

Model	Method	DINO ↑	CLIP-T ↑	CLIP-I ↑	Avg.
SD-v1.4	DreamBooth	0.679	0.323	0.801	0.601
	Textual Inversion	0.649	0.313	0.801	0.588
	LoLDU	0.723	0.319	0.830	0.750

Table VI. Comparison of Image Generation Methods. Performance metrics (DINO, CLIP-T, and CLIP-I) for DreamBooth, Textual Inversion, and LoLDU methods. Higher values indicate better performance. Bold values indicate best performance for each metric.

c) Results

We present the visual results in Figure 6, while Table VI provides a quantitative comparison. We assess our method's efficacy through DINO, CLIP-T and CLIP-I metrics. DINO^[66] is computed as the average pairwise cosine similarity between the ViT-S/16 DINO embeddings of generated and real images. CLIP-I measures the average pairwise cosine similarity between CLIP^[63] embeddings of generated and real images, while CLIP-T evaluates prompt fidelity by measuring the average cosine similarity between prompt and image CLIP embeddings. LoLDU achieves the highest average score across metrics.

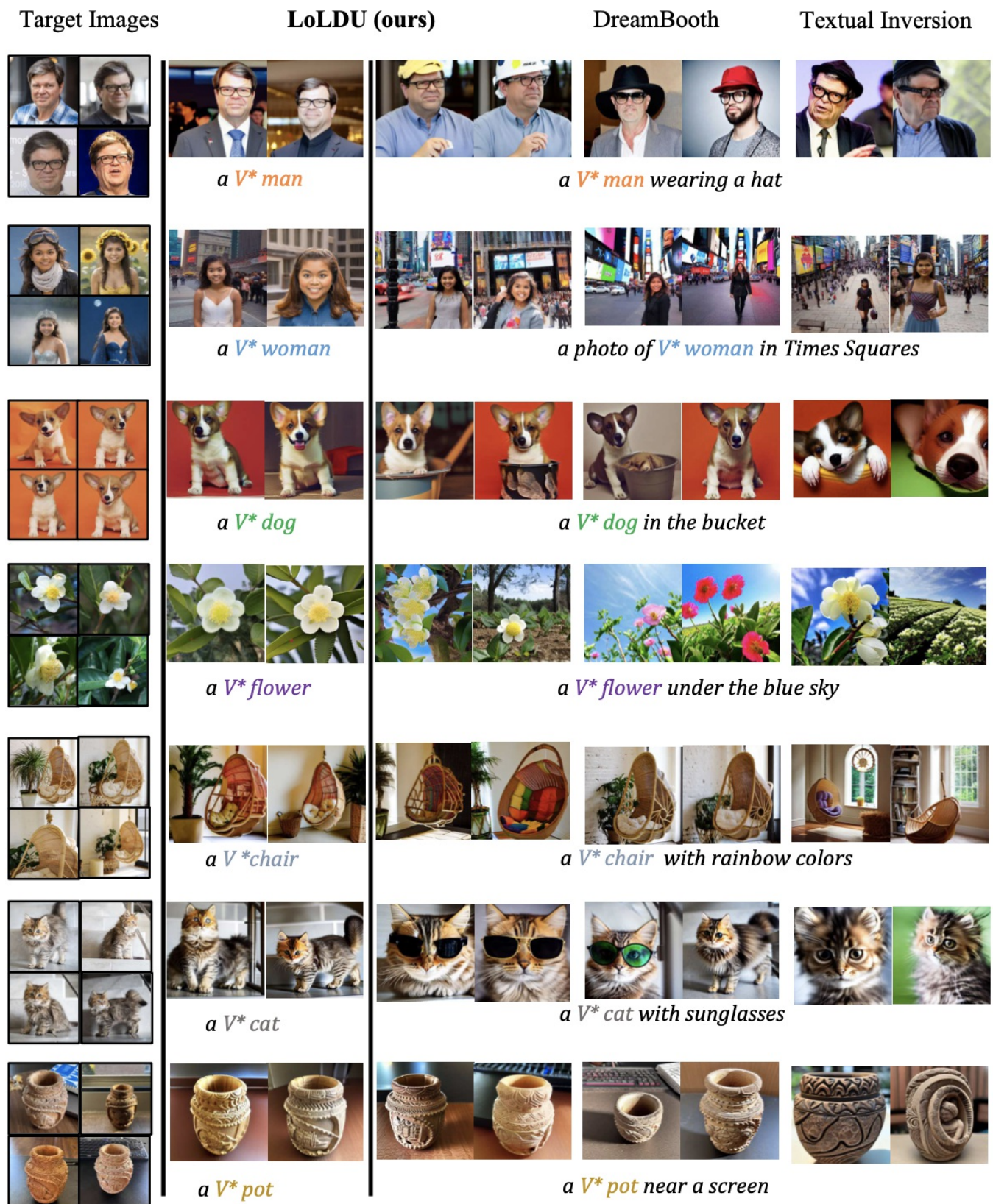


Figure 6. Visualized Results of the Image Generation Task. From left to right: target reference images, outputs from LoLDU (ours), DreamBooth, and Textual Inversion. Each row represents a distinct category with a specified prompt (annotated under each row). LoLDU demonstrates efficacy in generating diverse, prompt-adherent images while preserving key attributes from the reference set.

E. Analysis

In this section, we conduct a comprehensive analysis of the hyperparameters associated with LoLDU, specifically focusing on initialization, scaling factor, and rank. We systematically investigate the influence of these parameters on the performance and efficiency of our method across a variety of tasks.

a) Effect of Initialization

The initialization of the entries z in the diagonal matrix $diag(z)$ (Eq. 1) plays a crucial role in LoLDU's performance. We evaluate several initialization policies on the ViT Base model across six image classification datasets. Table V presents our findings.

Empirical results indicate that Uniform initialization consistently outperforms other strategies, achieving the highest average accuracy by stabilizing the training loop and enhancing convergence. Thus, LoLDU with Uniform initialization is optimal for applications requiring stable dynamics and high accuracy. Additionally, both Uniform and Normal initialization contribute to training stability.

b) Impact of Scaling Factor

The scaling factor within LoLDU is crucial for assessing the efficacy of low-rank updates in augmenting model performance. This ablation study is dedicated to examining the necessity of integrating a scaling factor, specifically fixed at a value of 1, to evaluate its impact on enhancing model accuracy and ensuring training stability.

Table IV presents a comprehensive comparative analysis of performance metrics with and without the incorporation of a scaling factor across various datasets. The empirical findings reveal that the absence of a scaling factor, as denoted by the gray values, consistently leads to diminished accuracy and compromises the stability of the convergence process. This highlights the pivotal role of the scaling factor in optimizing the performance of LoLDU, thereby enabling robust and efficient learning dynamics across a diverse range of image classification tasks.

c) Influence of Rank

The rank parameter within LoLDU is pivotal in determining the model's complexity and expressiveness. We conducted an extensive analysis by varying the rank across diverse tasks, as detailed in Table XII. Additionally, the visual results of this analysis are presented in Figure 4.

Our findings indicate that an increase in rank consistently enhances performance across all datasets, especially at lower ranks, but stabilizes beyond 256, indicating diminishing returns. Thus, selecting an optimal rank balances expressiveness and efficiency. In practical applications of LoLDU, our findings suggest that adopting a rank approximately one-third of the full rank ensures an optimal balance between performance and resource efficiency, thereby providing broader applicability across various scenarios.

d) *Parameter Efficiency vs. Performance Trade-off*

Finally, we explore the nuanced relationship between parameter efficiency and performance, focusing on the capabilities of LoLDU in comparison to other established methodologies.

Table II provides a compelling insight into the efficiency of LoLDU, which achieves a mean accuracy of 82.79% while utilizing a mere 0.21% of the parameters. This is a stark contrast to methods like FullFT, which, despite achieving a higher accuracy of 88.20%, require the full parameter set, and LoRA, which uses 6.77% of the parameters for a lower accuracy of 76.22%. These data underscore LoLDU's exceptional capacity to deliver competitive performance with a substantially reduced parameter footprint.

LoLDU's efficiency in parameter usage not only reduces computational and memory demands but also enhances the model's adaptability to various deployment scenarios, particularly those with limited resources. This efficiency is achieved without compromising on key performance metrics, as evidenced by the method's ability to maintain orthogonality, avoid random initialization, eliminate extra inference costs, and ensure faster convergence. These attributes collectively position LoLDU as a highly effective and resource-efficient alternative to traditional methods, offering a strategic advantage in both research and practical applications.

V. Conclusion

In conclusion, LoLDU represents a significant advancement in Parameter-Efficient Fine-Tuning (PEFT), offering a novel approach with the Lower-Diag-Upper (LDU) decomposition technique. By optimizing just 0.00025% of parameters while maintaining performance across diverse tasks and model architectures, LoLDU addresses the prohibitive computational and storage costs associated with fine-tuning large models. Its preservation of orthogonality in triangular matrices and precise diagonal matrix optimization ensure efficient scale transformation and robust convergence. Our extensive evaluation, spanning various tasks and model scales up to 7 billion parameters, validates LoLDU's effectiveness and superiority over traditional fine-tuning methods, underscoring its potential for broad applicability and impact in advancing efficient model customization practices.

Appendix

This appendix provides supplementary material to support the methodologies and findings presented in the main manuscript. It is organized into five key areas: Natural Language Understanding, Instruction Tuning, Image Classification, Image Generation, and Ablation Studies. Each section offers detailed insights into datasets, experimental protocols, and hyperparameter settings, ensuring the replicability and validation of our results.

- Section A: Analysis of the GLUE benchmark and hyperparameters for Natural Language Understanding tasks.
- Section B: Examination of the Alpaca dataset and LLaMA-2 model fine-tuning hyperparameters for Instruction Tuning.
- Section C: Overview of image classification datasets and Vision Transformer (ViT) fine-tuning configurations.

- Section D: Exploration of datasets for image generation and Stable Diffusion hyperparameters.
- Sections E: Ablation studies on learning rate and rank variations affecting model performance.

A. Natural Language Understanding

1. GLUE Benchmark Details

The GLUE benchmark is a framework for evaluating NLP models across nine tasks, such as CoLA, SST-2, and MRPC, focusing on grammaticality, sentiment, and semantic similarity. It includes a diagnostic dataset for assessing linguistic phenomena, aiding in the development of robust NLP systems through transfer learning. For more details, see the [GLUE Benchmark Overview](#).

2. Hyperparameters for GLUE Experiments

Table VII details the hyperparameters for GLUE experiments.

Table VII. Hyperparameters for GLUE Tasks			
Task	LR	Epochs	Max Length
MNLI	3e-4	10	128
SST-2	4e-4	10	128
MRPC	3e-4	20	512
CoLA	2e-4	20	128
QNLI	2e-4	10	512
QQP	3e-4	20	512
RTE	4e-4	20	512
STS-B	2e-4	30	512

Base: roberta-base, Batch: 32, Rank: 768, Alpha: 768

Modules: query, value, Warmup: 0.06

B. Instruction Tuning

1. Alpaca Dataset Overview

The Alpaca dataset serves as a crucial asset for instruction tuning, consisting of 52,000 instruction-output pairs generated using OpenAI's 'text-davinci-003' engine. Its primary goal is to improve the instruction-following capabilities of language models by providing a diverse array of instructional scenarios. The dataset is produced through the Self-Instruct framework, which includes modifications such as employing 'text-davinci-003' for instruction generation and implementing aggressive batch decoding to enhance efficiency. The Alpaca dataset's diversity and high-quality annotations make it a

valuable resource for training models to perform well across various tasks. This section explores the distinctive features of the Alpaca dataset, highlighting its role in the fine-tuning process of language models. For more details, refer to the [Hugging Face dataset card for Alpaca](#).

2. *Hyperparameters for LLaMA-2 Fine-tuning*

Table VIII provides a comprehensive overview of the hyperparameter settings employed during the fine-tuning of the LLaMA-2 model. These parameters are critical for optimizing model performance and ensuring robust convergence across various tasks.

Table VIII. Hyperparameters for Instruction Tuning	
Hyperparameter	Value
Base Model	LLaMA2-7B
Precision	BF16
Batch Size	128
Micro Batch Size	1
Learning Rate	1e-3
Number of Epochs	3
Rank	1024
Alpha	1024
Target Modules	q_proj, v_proj
Cutoff Length	256
Seed	42

C. *Image Classification*

1. *Dataset Descriptions*

This section introduces the datasets employed for image classification tasks, which include CIFAR10^[25], EuroSAT^[30], StanfordCars^[28], FLOWERS102^[27], FGVC^[29], and CIFAR100^[26]. These datasets are selected to represent a broad spectrum of visual concepts and complexities, ranging from small to large label spaces.

2. *Hyperparameters for ViT Fine-tuning*

The hyperparameter settings utilized for the fine-tuning of the Vision Transformer (ViT) model are detailed in Table IX.

Table IX. Hyperparameters for Image Classification

Hyperparameter	Value
Model	vit-b16-224-in21k
Learning Rate	3e-3
Batch Size	128
Max Epochs	10
Precision	bf16
Optimizer	AdamW
LR Scheduler	Linear
Warmup Steps	30
Target Modules	query, value
Rank	768
Alpha	768
Seed	42

D. Image Generation

1. Dataset Details

The CustomConcept101 and Flickr-Faces-HQ (FFHQ) datasets provide concept images for fine tuning our image generation model. FFHQ contains 70,000 high-resolution images (1024×1024) with diverse attributes such as age, ethnicity, and accessories. Images were sourced from Flickr, aligned, and cropped using dlib, excluding non-human subjects. For more information, see the [FFHQ Dataset](#).

2. Hyperparameters for Stable Diffusion Fine-tuning

The hyperparameter settings utilized for the fine-tuning of the Stable Diffusion model are detailed in Table X.

Table X. Hyperparameters for Image Generation

Hyperparameter	Value
Base Model	stable-diffusion-v1-5
VAE	sd-vae-ft-mse
Learning Rate	5e-4
Precision	fp16
Resolution	512
Train Batch Size	1
Optimizer	AdamW
LR Scheduler	constant
LR Warmup Steps	15
Max Train Steps	1000
Rank	32
Alpha	32
Seed	42
Adam Weight Decay	0.01
Target Modules	to_k, to_v, to_q, to_out

E. Ablation Studies

1. Learning Rate

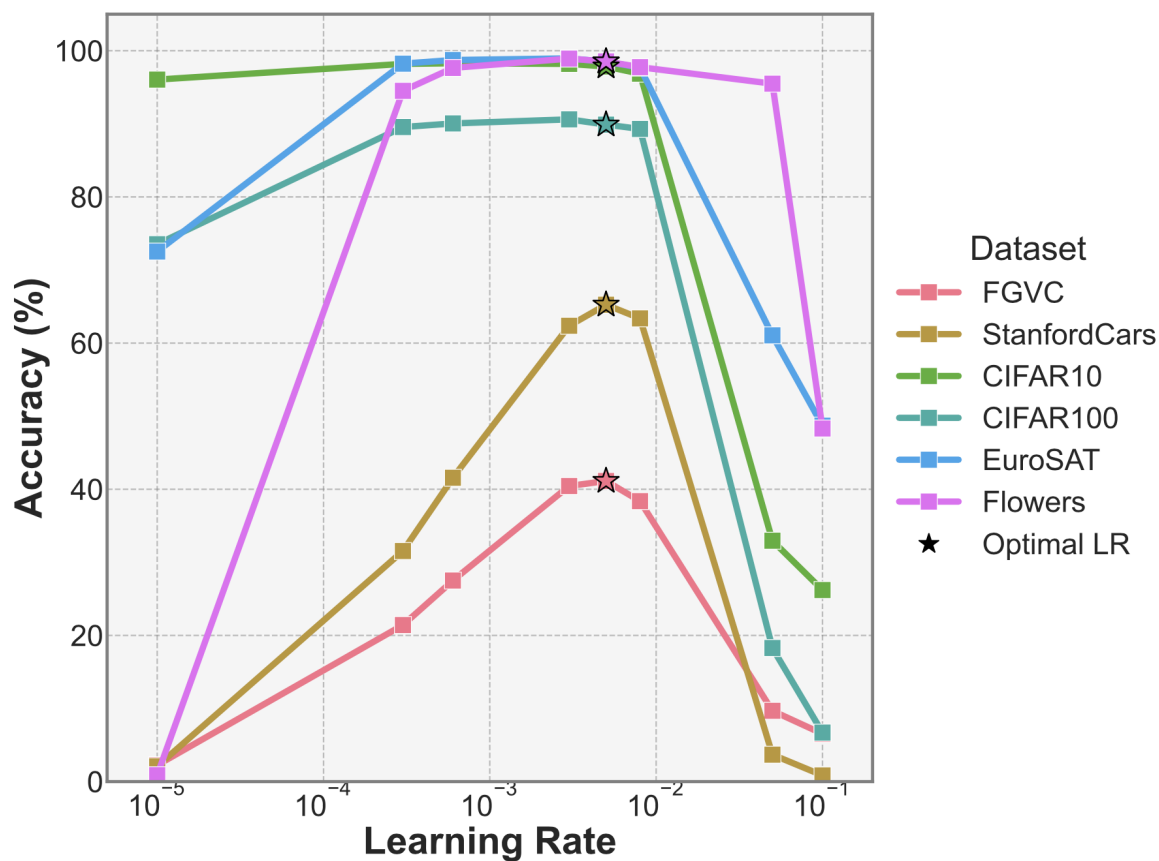


Figure 7. Learning Rate Ablation Study. The figure demonstrates the effect of different learning rates on ViT-base model accuracy across FGVC, StanfordCars, CIFAR10, CIFAR100, EuroSAT, and Flowers datasets.

LR	FGVC acc	StanfordCars acc	CIFAR10 acc	CIFAR100 acc	EuroSAT acc	Flowers acc	Avg.
ViT-Base LR Ablation							
1e-1	6.54	0.85	26.21	6.71	48.70	48.31	22.89
5e-2	9.69	3.69	32.96	18.28	61.06	95.49	36.86
8e-3	38.37	63.38	96.86	89.30	97.69	97.75	80.56
5e-3	41.13	65.25	97.84	89.89	98.50	98.53	81.86
3e-3	40.44	62.37	98.20	90.61	98.95	98.92	81.58
6e-4	27.51	41.57	98.28	90.05	98.73	97.65	75.63
3e-4	21.42	31.55	98.20	89.56	98.23	94.51	72.25
1e-5	2.25	2.10	96.05	73.53	72.53	0.88	41.22

Table XI. LR Ablation for ViT-Base: Comparison on FGVC, StanfordCars, CIFAR10, CIFAR100, EuroSAT, and Flowers. All ranks set to 768. Bold indicates best results.

This section provides an academic analysis of the impact of varying learning rates on model training. The visual representation, as detailed in 7, illustrates the outcomes of the learning rate ablation study, while the accompanying table,

referenced in XI, provides comprehensive quantitative data.

2. Rank Ablation

Rank	Params	FGVC	StanfordCars	CIFAR10	CIFAR100	EuroSAT	Flowers
ViT-Base Rank Ablation							
1	24	27.59	43.95	96.81	86.67	95.25	98.33
8	192	28.28	48.40	97.47	89.84	96.14	98.53
16	384	31.13	50.87	97.76	88.48	96.74	98.53
32	768	32.75	53.00	97.82	88.76	97.28	98.63
64	1536	34.01	55.09	97.96	89.60	97.60	98.53
128	3072	34.91	58.20	98.07	89.89	98.20	98.53
256	6144	36.38	61.44	98.06	90.18	98.62	98.63
512	12288	38.48	63.68	98.17	90.30	98.83	98.63
768	18456	42.15	66.66	98.59	91.21	99.21	98.63
FT	85.8	54.84	79.78	98.92	92.38	99.05	98.43
LoRA	581	25.16	45.38	98.78	92.02	98.44	97.55

Table XII. ViT Rank Ablation Study on FGVC, StanfordCars, CIFAR10, CIFAR100, EuroSAT, and Flowers datasets. Different ranks indicate varying parameter counts. #Params: Tunable parameters (M). The first section shows the base version, followed by the large-scale ablation. Bold denotes optimal LoLDU results.

This subsection presents an analysis of the rank ablation study, examining the impact of different parameter ranks on model performance. Table XII summarizes the results.

Footnotes

¹ Kindly note that the parameter count reported does not include the classification head, as it must be trained using all methods.

References

1. [^]Liu H, Li C, Wu Q, Lee YJ. "Visual instruction tuning." In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. 2023.

2. [^]Zhang L, Wei W, Shi Q, et al. "Accurate tensor completion via adaptive low-rank representation." *IEEE Transactions on Neural Networks and Learning Systems*. 2020;(10).

3. ^aZhang L, Fu J, Wang S, et al. "Guide subspace learning for unsupervised domain adaptation." *IEEE Transactions on Neural Networks and Learning Systems*. 2020;(9).
4. ^aZhang R, Zhang H, Li X, Nie F. "Adaptive robust low-rank 2-d reconstruction with steerable sparsity." *IEEE Transactions on Neural Networks and Learning Systems*. 2020;(9).
5. ^{a, b, c, d}Gal R, Alaluf Y, Atzmon Y, et al. "An image is worth one word: Personalizing text-to-image generation using textual inversion." In: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. 2023.
6. ^{a, b, c, d, e, f}Ruiz N, Li Y, Jampani V, et al. "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. 2023.
7. ^aLai S, Liu C, Wang D, Lu H. "Refocus the attention for parameter-efficient thermal infrared object tracking." *IEEE Transactions on Neural Networks and Learning Systems*. 2024.
8. ^aZheng Z, Wang X, Zheng N, Yang Y. "Parameter-efficient person re-identification in the 3d space." *IEEE Transactions on Neural Networks and Learning Systems*. 2024;(6).
9. ^{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p}Hu EJ, Shen Y, Wallis P, et al. "Lora: Low-rank adaptation of large language models." In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. 2022.
10. ^{a, b, c, d, e}Kopiczko DJ, Blankevoort T, Asano YM. "VeRA: Vector-based random matrix adaptation." In: *The Twelfth International Conference on Learning Representations*. 2024.
11. ^aDettmers T, Pagnoni A, Holtzman A, Zettlemoyer L. "Qlora: Efficient finetuning of quantized llms." In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. 2023.
12. ^{a, b, c, d}Meng F, Wang Z, Zhang M. "PiSSA: Principal Singular Values and Singular Vectors Adaptation of Large Language Models." 2024.
13. ^{a, b, c}Chen X, Xie S, He K. "An empirical study of training self-supervised vision transformers." In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. 2021.
14. ^{a, b, c, d}Gao Z, Wang Q, Chen A, et al. "Parameter-Efficient Fine-Tuning with Discrete Fourier Transform." 2024.
15. ^{a, b, c, d}Touvron H, Martin L, Stone K, et al. "Llama 2: Open Foundation and Fine-Tuned Chat Models." 2023.
16. ^{a, b, c, d, e}Taori R, Gulrajani I, Zhang T, et al. "Stanford alpaca: An instruction-following llama model." 2023.
17. ^{a, b, c}Chiang WL, Li Z, Lin Z, et al. "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality." 2023.
18. ^{a, b, c}Jiang T, Huang S, Luo S, et al. "MoRA: High-Rank Updating for Parameter-Efficient Fine-Tuning." 2024.
19. ^{a, b}Glorot X, Bengio Y. "Understanding the difficulty of training deep feedforward neural networks." In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 2010.
20. ^{a, b}Sutskever I, Martens J, Dahl GE, Hinton GE. "On the importance of initialization and momentum in deep learning." In: *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*. 2013.

21. ^{a, b, c} Qiu Z, Liu W, Feng H, et al. "Controlling text-to-image diffusion by orthogonal finetuning." In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. 2023.
22. ^{a, b} Liu W, Lin R, Liu Z, et al. "Orthogonal over-parameterized training." In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. 2021.
23. ^{a, b} Chia YK, Hong P, Bing L, Poria S. "InstructEval: Towards holistic evaluation of instruction-tuned large language models." In: *Proceedings of the First edition of the Workshop on the Scaling Behavior of Large Language Models (SCALE-LLM 2024)*. 2024.
24. ^{a, b, c} Wang A, Singh A, Michael J, et al. "GLUE: A multi-task benchmark and analysis platform for natural language understanding." In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 2019.
25. ^{a, b, c, d} Krizhevsky A, Nair V, Hinton G. "CIFAR-10 (Canadian Institute for Advanced Research)."
26. ^{a, b, c, d} CIFAR-100 (Canadian Institute for Advanced Research).
27. ^{a, b, c, d} Gurnani A, Mavani V, Gajjar V, Khandhediya Y. "Flower Categorization using Deep Convolutional Neural Networks." 2017.
28. ^{a, b, c, d} Krause J, Stark M, Deng J, Fei-Fei L. "3D Object Representations for Fine-Grained Categorization." In: *2013 IEEE International Conference on Computer Vision Workshops*. 2013.
29. ^{a, b, c, d} Maji S, Rahtu E, Kannala J, et al. "Fine-Grained Visual Classification of Aircraft." 2013.
30. ^{a, b, c, d} Helber P, Bischke B, Dengel A, Borth D. "EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification." 2017.
31. ^{a, b, c} Y. Liu, M. Ott, N. Goyal, and et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," 2019.
32. ^{a, b, c} A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.
33. ^{a, b, c} R. Rombach, A. Blattmann, D. Lorenz, and et al., "High-resolution image synthesis with latent diffusion models," 2021.
34. [^] N. Houlsby, A. Giurghi, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, 2019*.
35. [^] T. Lei, J. Bai, S. Brahma, J. Ainslie, K. Lee, Y. Zhou, N. Du, V. Y. Zhao, Y. Wu, B. Li, Y. Zhang, and M. Chang, "Conditional adapters: Parameter-efficient transfer learning with fast inference," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
36. [^] R. Zhang, J. Han, C. Liu, and et al., "LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init

Attention," 2023.

37. ^{a, b}F. Zhang and M. Pilanci, "Spectral Adapter: Fine-Tuning in Spectral Space," 2024.
38. [^]X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," 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)*, 2021.
39. [^]D. Guo, A. Rush, and Y. Kim, "Parameter-efficient transfer learning with diff pruning," 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)*, 2021.
40. [^]S. S. S. Das, H. Zhang, P. Shi, W. Yin, and R. Zhang, "Unified low-resource sequence labeling by sample-aware dynamic sparse finetuning," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.
41. [^]A. Ansell, I. Vulic, H. Sterz, and et al., "Scaling Sparse Fine-Tuning to Large Language Models," 2024.
42. [^]Y. Sung, V. Nair, and C. Raffel, "Training neural networks with fixed sparse masks," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021.
43. [^]E. Ben Zaken, Y. Goldberg, and S. 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)*, 2022.
44. [^]S.-Y. Liu, C.-Y. Wang, H. Yin, and et al., "DoRA: Weight-Decomposed Low-Rank Adaptation," in *Proceedings of the 41st International Conference on Machine Learning*, July 2024.
45. [^]D. Vander Mijnsbrugge, F. Ongenae, and S. Van Hoecke, "Parameter efficient neural networks with singular value decomposed kernels," *IEEE Transactions on Neural Networks and Learning Systems*, no. 9, 2023.
46. ^{a, b}A. Aghajanyan, S. Gupta, and L. 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)*, 2021.
47. [^]J. Phang, Y. Mao, P. He, and W. Chen, "Hypertuning: Toward adapting large language models without back-propagation," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, 2023*.
48. [^]C. Feng, M. He, Q. Tian, and et al., "TriLoRA: Integrating SVD for Advanced Style Personalization in Text-to-Image Generation," 2024.
49. ^{a, b}V. Lingam, A. Tejaswi, A. Vavre, and et al., "SVFT: Parameter-Efficient Fine-Tuning with Singular Vectors," 2024.
50. ^{a, b}L. Han, Y. Li, H. Zhang, P. Milanfar, D. N. Metaxas, and F. Yang, "Svdif: Compact parameter space for diffusion fine-tuning," in *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023, 2023*.
51. [^]T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*,

NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023.

52. ^{a, b, c} Q. Zhang, M. Chen, A. Bukharin, and et al., "AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning," 2023.
53. [^] C. Li, H. Farkhoor, R. Liu, and J. Yosinski, "Measuring the intrinsic dimension of objective landscapes," in 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018.
54. [^] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.
55. [^] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural networks for machine learning, no. 2, 2012.
56. [^] N. Ding, X. Lv, Q. Wang, Y. Chen, B. Zhou, Z. Liu, and M. Sun, "Sparse low-rank adaptation of pre-trained language models," in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023.
57. ^{a, b} P. Ren, C. Shi, S. Wu, M. Zhang, Z. Ren, M. Rijke, Z. Chen, and J. Pei, "MELoRA: Mini-ensemble low-rank adapters for parameter-efficient fine-tuning," in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), August 2024.
58. [^] M. Valipour, M. Rezagholizadeh, I. Kobyzev, and A. 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, 2023.
59. [^] S. Longpre, L. Hou, T. Vu, A. Webson, H. W. Chung, Y. Tay, D. Zhou, Q. V. Le, B. Zoph, J. Wei, and A. Roberts, "The flan collection: Designing data and methods for effective instruction tuning," in International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, 2023.
60. [^] A. Köpf, Y. Kilcher, D. von Rütte, S. Anagnostidis, Z. R. Tam, K. Stevens, A. Barhoum, D. Nguyen, O. Stanley, R. Nagyfi, S. ES, S. Suri, D. Glushkov, A. Dantuluri, A. Maguire, C. Schuhmann, H. Nguyen, and A. Mattick, "Openassistant conversations - democratizing large language model alignment," in Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023.
61. [^] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "ImageNet-21K Pretraining for the Masses," 2021.
62. [^] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, "LAION-5B: an open large-scale dataset for training next generation image-text models," in Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022.
63. ^{a, b} A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, 2021.
64. [^] N. Kumari, B. Zhang, R. Zhang, E. Shechtman, and J. Zhu, "Multi-concept customization of text-to-image diffusion," in

IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023, 2023.

65. [^] T. Karras, S. Laine, and T. Aila, "Flickr faces hq (ffhq) 70k from stylegan," *CoRR*, 2018.
66. [^] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, 2021.*