Beyond Decoupled PEFT: Geometry-Aware Low-Rank Adaptation via Riemannian Reparameterization

Yuanyang Cao[1], **Xichun Liu**[1], **Haitao Jiang**[1], **Jianji Wang**[1]*

[1]State Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China

caoyuan1418@stu.xjtu.edu.cn, liuxichun@stu.xjtu.edu.cn
jianghaitao@stu.xjtu.edu.cn, wangjianji@mail.xjtu.edu.cn

Abstract

Low-Rank Adaptation (LoRA) and its weight-decomposed variants dominate parameter-efficient fine-tuning, yet their directional update mechanisms remain geometrically mismatched to the normalized parameterization they impose. In existing decoupled PEFT methods, low-rank directional updates are computed in unconstrained Euclidean space and only then projected back onto a spherical manifold, which can introduce capacity redundancy and directional gradient distortion. To address these limitations, we propose Geodesic Orthogonal Low-Rank Adaptation (GO-LoRA), a geometry-aware reparameterization framework for decoupled PEFT. GO-LoRA first applies orthogonal tangent projection to remove components parallel to the pretrained base direction, ensuring that the low-rank directional budget is devoted entirely to tangent-space motion. It then uses a geodesic flow based on the Riemannian Exponential Map to construct manifold-consistent directional updates on the hypersphere, improving geometric fidelity relative to Euclidean normalization while retaining standard optimization over the underlying low-rank parameters. Extensive experiments across language, vision, code generation, and multimodal reasoning tasks show that GO-LoRA consistently outperforms strong LoRA baselines while introducing no additional inference-time overhead after offline merging. The code is available at `https://github.com/E-green11/go-lora`.

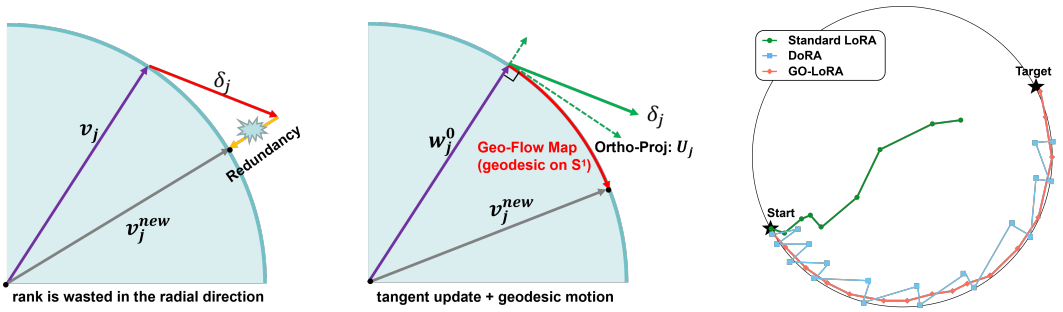
1 Introduction

The rapid advancement of Large Language Models (LLMs) [10] and Vision-Language Models (VLMs) [58, 46] is accompanied by prohibitive memory and computational costs during full-parameter fine-tuning (Full FT) for downstream tasks [40]. To alleviate this training bottleneck, Parameter-Efficient Fine-Tuning (PEFT) [25], most notably Low-Rank Adaptation (LoRA) [33], has emerged as the standard paradigm. By freezing pre-trained weights and optimizing strictly low-rank adapters, LoRA achieves competitive performance while significantly reducing trainable parameters. Nonetheless, recent studies show that LoRA still exhibits an expressiveness gap relative to FT, particularly on complex reasoning and multimodal tasks [2].

To bridge this gap, recent advancements have explored structural re-parameterizations and weight-decomposed adaptations [50, 53, 78, 41, 48]. These methods aim to better match the behavior of FT by separating pre-trained weights into magnitude and direction components, or by reallocating singular subspaces. However, despite this structural decoupling, their directional updates are computed in an unconstrained Euclidean space before being projected back onto a spherical manifold.

*Corresponding author.

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



(a) Standard Decoupled PEFT (b) GO-LoRA (Ours) (c) Stochastic Trajectories on $\mathcal{S}^1$

Figure 1: Comparison of geometric update mechanisms between standard decoupled PEFT and GO-LoRA. (a) Standard decoupled PEFT: an unconstrained update around the current directional variable v_j contains a radial component that is squandered by subsequent L_2 normalization, leading to capacity redundancy. (b) GO-LoRA: relative to the frozen pretrained anchor w_j^0, the update is first projected onto the tangent space (U_j) and then mapped by Geo-Flow to produce a geodesic-consistent directional parameterization. (c) Illustrative 2D dynamics under a shared noisy target-seeking signal: LoRA exhibits unconstrained Euclidean drift, DoRA follows secant-style normalized updates, and GO-LoRA follows a smoother trajectory that approaches the target more directly on $\mathcal{S}^1$.

This geometric mismatch gives rise to two bottlenecks, illustrated in Figure 1: (1) capacity redundancy, where part of the low-rank update budget is spent on radial components that are subsequently removed by normalization; and (2) suboptimal optimization trajectories, where naive Euclidean additions trace secant-style approximations rather than the intrinsic manifold geometry, leading to less faithful gradient flow. To provide an intuitive illustration, our 2D simulation suggests that these bottlenecks can produce more oscillatory and less coherent trajectories under noisy gradients.

To address these issues, we propose Geodesic Orthogonal Low-Rank Adaptation (GO-LoRA), a geometry-aware reparameterization framework for decoupled PEFT. Rather than performing state-space Riemannian optimization over a moving manifold point, GO-LoRA uses the frozen pretrained weight as an anchor, learns low-rank tangent-space updates relative to this anchor, and maps them to the hypersphere through the Riemannian Exponential Map. Concretely, GO-LoRA introduces two parameter-free mechanisms. First, Orthogonal Tangent Projection (Ortho-Proj) projects the low-rank update strictly onto the tangent space of the pretrained base weight, removing redundant parallel components so that the directional update budget is devoted entirely to tangent-space motion. Second, Geodesic Flow (Geo-Flow) replaces the heuristic secant-style update with a geodesic construction induced by the Exponential Map, yielding a smooth and norm-preserving manifold-consistent directional parameterization. These two mechanisms are intrinsically coupled: Geo-Flow requires a valid tangent direction, which is exactly supplied by Ortho-Proj. Together, they provide a geometrically coherent directional branch for low-rank adaptation while retaining standard optimization over the underlying low-rank parameters.

In summary, our main contributions are threefold:

- **Geometric Diagnosis:** We identify a geometric limitation in existing weight-decomposed PEFT methods. We show that unconstrained Euclidean directional updates induce capacity redundancy and secant-style directional gradient distortion after post-hoc normalization, thereby reducing directional efficiency and weakening optimization dynamics.

- **The GO-LoRA Framework:** We propose GO-LoRA, a geometry-aware reparameterization framework for decoupled PEFT. By integrating *Orthogonal Tangent Projection* and *Geodesic Flow*, GO-LoRA removes redundant radial components in the directional branch and replaces heuristic secant-style updates with manifold-consistent geodesic parameterization, without introducing additional inference-time adaptation overhead after offline merging.

- **Strong Empirical Gains:** Extensive experiments across vision, language, code generation, and multimodal reasoning tasks show that GO-LoRA consistently outperforms the latest strong PEFT baselines, with especially clear gains under tight rank budgets.

2 Related Work

Parameter-Efficient Fine-Tuning. As foundation models scale, full FT becomes increasingly expensive. PEFT mitigates this by updating only a small fraction of the model space [32, 47, 44, 75]. Beyond early adapter- and prompt-based strategies [37, 14], Low-Rank Adaptation (LoRA) has motivated a broad family of variants, including methods for quantization [24], dynamic rank allocation [78, 67], random projection [41], modified learning rates [29, 68], rank stabilization [39], and long-context scaling [15]. To narrow the gap toward full FT, recent works explore decoupled re-parameterizations: DoRA [50] isolates magnitude from directional updates, while others optimize spectral or orthogonal initialization [53, 11], explore Stiefel subspaces [48], Hadamard high-rank updates [35], or share weights across layers [59]. A common feature is that directional updates are computed in unconstrained Euclidean space before normalization, which can leave redundant parallel components in the normalized directional branch and limit low-rank budget efficiency.

Orthogonality Constraints in Deep Learning. Orthogonality has long been used to stabilize gradient flow, prevent collapse, and preserve high effective rank in learned representations [63, 5, 57, 56]. Techniques ranging from strict orthogonal initialization to orthogonality-preserving normalization have been shown to preserve representation quality and rank structure [34]. Recent PEFT frameworks have likewise introduced orthogonality constraints [51]. Nevertheless, the explicit use of deterministic Gram-Schmidt-style projection to remove the component of a low-rank update parallel to a pretrained base direction, thereby confining the effective update to the tangent space of that direction, has received limited attention in decoupled PEFT.

Riemannian Geometry and Manifold-Aware Optimization. Optimization on curved spaces, particularly hyperspheres and orthogonal manifolds, provides principled tools for maintaining structural constraints such as norm preservation [1, 8, 9, 43]. Riemannian methods have been incorporated into deep learning for hierarchical embeddings, constrained representations, and adaptive manifold-aware optimizers [18, 6, 45]. Our setting is different: GO-LoRA does not perform classical state-space Riemannian optimization over a moving manifold variable. Instead, it uses the pretrained weight as a fixed anchor, projects low-rank updates into the corresponding tangent space, and maps them to the hypersphere through the Exponential Map. This brings manifold-aware parameterization and exact geodesic construction into decoupled low-rank adaptation while retaining Euclidean optimization over the low-rank factors, without introducing manifold solvers or trainable parameters.

3 Method

In this section, we formally introduce the GO-LoRA framework. We first revisit the standard weight decomposition paradigm and mathematically characterize two critical bottlenecks: capacity redundancy and suboptimal optimization trajectories. We then recast the directional branch through the lens of Riemannian geometry and present the two core mechanisms of GO-LoRA that address these flaws in a geometrically coherent manner: Ortho-Proj and Geo-Flow.

3.1 Revisiting Decoupled PEFT and its Geometric Flaws

For a pre-trained linear layer with weight matrix $W_0 \in \mathbb{R}^{d\times k}$, standard LoRA introduces a low-rank update $\Delta W = BA$, where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ with intrinsic rank $r \ll \min(d, k)$. To bridge the expressiveness gap between LoRA and full FT, weight-decomposed approaches [50] decouple the pre-trained weight W_0 into a trainable magnitude vector $m \in \mathbb{R}^{1\times k}$ and a directional branch parameterized by a matrix $V \in \mathbb{R}^{d\times k}$, typically initialized from W_0. The standard decoupled forward pass is functionally defined as:

$$W = m \odot \frac{V + BA}{\|V + BA\|_c}, \tag{1}$$

where $\|\cdot\|_c$ denotes the column-wise L_2 norm, and $\odot$ denotes broadcasted scalar multiplication over the respective columns. Although different decoupled variants instantiate the directional branch somewhat differently, they all share the same essential pattern: an unconstrained directional update is formed in ambient Euclidean space and only then normalized back onto a sphere.

Let $v_j \in \mathbb{R}^d$ and $\delta_j = (BA)_{:,j} \in \mathbb{R}^d$ denote the j-th column of V and the unconstrained low-rank update BA, respectively. The normalized direction update implemented in the architecture is computed as $f(v_j, \delta_j) = \frac{v_j+\delta_j}{\|v_j+\delta_j\|_2}$.

The Capacity Redundancy Paradox. Because the unconstrained update vector δ_j is optimized in Euclidean space $\mathbb{R}^d$, it generally contains a component parallel to the current base direction v_j. Let us orthogonally decompose δ_j into $\delta_j = \alpha v_j + \delta_j^\perp$, where αv_j is the parallel collinear component and $\delta_j^\perp$ is orthogonal to v_j. Assuming the typical fine-tuning regime where the update perturbation is relatively small ($\alpha > -1$), substituting this decomposition into the normalization function yields:

$$f(v_j, \delta_j) = \frac{v_j + \alpha v_j + \delta_j^\perp}{\|(1+\alpha)v_j + \delta_j^\perp\|_2} = \frac{v_j + \frac{1}{1+\alpha}\delta_j^\perp}{\sqrt{\|v_j\|_2^2 + \frac{1}{(1+\alpha)^2}\|\delta_j^\perp\|_2^2}}. \tag{2}$$

Equation 2 reveals a critical limitation of prior decoupled methods: after normalization, the learned collinear factor α acts only as a scalar rescaling of the orthogonal component $\delta_j^\perp$. In other words, learning a non-zero α is functionally equivalent, in the normalized directional branch, to setting $\alpha = 0$ and replacing $\delta_j^\perp$ with the rescaled update $\hat{\delta}_j^\perp = \frac{1}{1+\alpha}\delta_j^\perp$. Therefore, the parallel component αv_j contributes no independent directional degree of freedom on the hypersphere. Any portion of the low-rank update budget allocated to this collinear component is not used to induce new rotational motion, but is instead absorbed by the subsequent L_2 normalization. We refer to this structural inefficiency as capacity redundancy; Proposition 1 in Appendix A formalizes this statement. Importantly, this redundancy is structural rather than a guarantee of optimization inefficiency. Although the parallel component introduces no independent directional degree of freedom, it can rescale the effective tangent update and interact with the learnable magnitude branch during training. Proposition 1 therefore establishes representational redundancy, while its practical impact on optimization is assessed empirically.

The Secant Trajectory and Directional Gradient Distortion. Beyond spatial capacity waste, naive Euclidean addition also degrades the backward pass. Even if the parallel component were hypothetically removed, the decoupled method would still compute the new direction through a linear secant step followed by projection back onto the sphere: $f(v_j, \delta_j) = \frac{v_j+\delta_j}{\|v_j+\delta_j\|_2}$. During backpropagation, applying the chain rule to this L_2 normalization yields the Jacobian

$$\left.\frac{\partial f}{\partial \delta}\right|_{x=v_j+\delta_j} = \frac{1}{\|v_j+\delta_j\|_2}\left(I - \hat{x}_j\hat{x}_j^T\right), \quad \text{where } \hat{x}_j = \frac{v_j+\delta_j}{\|v_j+\delta_j\|_2}. \tag{3}$$

Here, $(I - \hat{x}_j\hat{x}_j^T)$ is an orthogonal projector with eigenvalues in $\{0, 1\}$. Therefore, the Jacobian has operator norm $\frac{1}{\|v_j+\delta_j\|_2}$, and the transmitted gradient in the normalized directional branch is uniformly scaled by at most this factor, while the radial component is removed by the projector. We note that, in the full decoupled parameterization, this directional branch is subsequently rescaled by the magnitude term m_j; accordingly, the analysis here should be interpreted as characterizing the backward geometry of the normalized directional branch itself, rather than as a standalone claim about the absolute gradient magnitude of the fully recomposed weight in every training regime. We refer to this optimization penalty as directional gradient distortion under post-hoc normalization; its formal quantification is given in Appendix D. Together, capacity redundancy restricts spatial expressiveness, while post-hoc normalization distorts the effective directional gradient pathway.

3.2 Riemannian Tangent Space and Orthogonal Projection

The preceding subsection analyzes the geometric limitations of generic decoupled PEFT through the directional variable V. GO-LoRA departs from this formulation by anchoring its directional parameterization at the frozen pretrained weight W_0, rather than treating the manifold reference point itself as a trainable state. For each column j, let $w_j^0 = W_{0[:,j]} \in \mathbb{R}^d$ denote the frozen base direction. The corresponding spherical manifold is

$$\mathcal{M}_j = \{x \in \mathbb{R}^d \mid \|x\|_2 = \|w_j^0\|_2\}, \tag{4}$$

with tangent space

$$T_{w_j^0}\mathcal{M}_j = \{U \in \mathbb{R}^d \mid \langle U, w_j^0\rangle = 0\}. \tag{5}$$

Given the unconstrained low-rank update $\delta_j = (BA)_{:,j}$, GO-LoRA applies Orthogonal Tangent Projection (Ortho-Proj) with respect to this frozen base direction:

$$U_j = \delta_j - \frac{\langle w_j^0, \delta_j \rangle}{\|w_j^0\|_2^2} w_j^0. \tag{6}$$

By construction, $\langle w_j^0, U_j \rangle = 0$, so $U_j \in T_{w_j^0}\mathcal{M}_j$. Thus, Ortho-Proj removes the component of δ_j parallel to the base direction and confines the directional update to tangent-space motion.

This explicit geometric constraint is fundamentally different from optimizer-level adaptation. Adaptive optimizers such as AdamW adjust parameter updates through coordinate-wise gradient statistics and may partially mitigate inefficient radial allocation during training. However, such preconditioning does not explicitly enforce orthogonality between the induced low-rank update and the pretrained weight direction. Consequently, even with adaptive optimization, the learned residual may retain a non-negligible component parallel to the directional reference. In contrast, Ortho-Proj operates directly on the effective low-rank update $\delta_j = (BA)_{:,j}$ and removes its component along the frozen anchor w_j^0 before constructing the normalized directional representation. This guarantees $\langle w_j^0, U_j \rangle = 0$ by construction, independently of the optimizer used to update A and B. Importantly, this constraint applies to the induced directional update rather than requiring the individual low-rank factors to satisfy manifold constraints. GO-LoRA therefore retains standard Euclidean optimization of its trainable parameters while ensuring that their combined directional contribution lies in the appropriate tangent space.

3.3 Geodesic Flow via the Exponential Map

GO-LoRA is not a state-space Riemannian optimizer that updates the manifold point itself at every step. Instead, it uses the frozen pretrained direction as a fixed anchor and learns tangent-space coordinates relative to this anchor, while the resulting directional branch is realized through the exponential map and differentiated through its exact Jacobian. While Ortho-Proj aligns the update with the tangent space, combining the frozen base direction and the tangent update through naive Euclidean addition still introduces a secondary geometric mismatch. In particular, the linear update $w_j^0 + U_j$ forms a secant-style approximation in ambient Euclidean space, rather than following the manifold-consistent geodesic induced by the spherical constraint. As the step size grows, this discrepancy becomes increasingly pronounced, leading to less faithful manifold navigation and weaker optimization dynamics.

To resolve this issue, GO-LoRA applies Geo-Flow based on the Riemannian Exponential Map [43]. Given the frozen base direction $w_j^0 = W_{0[:,j]}$ and the tangent update $U_j \in T_{w_j^0}\mathcal{M}_j$, the reparameterized directional vector is defined as

$$v_j^{\text{new}} = w_j^0 \cos(\theta_j) + \frac{U_j}{\|U_j\|_2} \|w_j^0\|_2 \sin(\theta_j), \qquad \theta_j = \frac{\|U_j\|_2}{\|w_j^0\|_2}. \tag{7}$$

This construction follows the geodesic induced by the exponential map on the sphere of radius $\|w_j^0\|_2$, replacing the secant-style Euclidean approximation with a manifold-consistent arc update.

For small angular displacements $\theta_j \to 0$, the Taylor expansion of Eq. 7 reduces to the first-order approximation $w_j^0 + U_j$. Therefore, Euclidean addition can be interpreted as a first-order local approximation to the geodesic update, whereas Geo-Flow preserves the correct manifold geometry at finite step sizes. In this sense, prior Euclidean updates can be viewed as a linearized surrogate of the exponential-map construction, with the discrepancy becoming more evident when larger directional adjustments are required.

Geo-Flow also preserves the spherical constraint exactly. Expanding the squared norm of Eq. 7 yields a cross term proportional to $\langle w_j^0, U_j \rangle$. Because Ortho-Proj guarantees $U_j \in T_{w_j^0}\mathcal{M}_j$, we have $\langle w_j^0, U_j \rangle = 0$, and the cross term vanishes. As a result, the update is exactly norm-preserving, i.e., $\|v_j^{\text{new}}\|_2 = \|w_j^0\|_2$, as proved in Appendix B. Thus, Ortho-Proj and Geo-Flow are intrinsically coupled: the tangent projection supplies a valid tangent direction, and the exponential map transports that direction along the sphere without requiring a post-hoc normalization step.

Beyond geometric fidelity in the forward pass, Geo-Flow also improves the gradient pathway relative to Euclidean normalization. Standard L_2 normalization introduces a secant-style tangent geometry in

the normalized directional branch and only removes radial components after the unconstrained update has already been formed. In contrast, GO-LoRA explicitly projects updates into the tangent space first, keeps the trigonometric factors inside the computation graph, and backpropagates through the exact Jacobian of the exponential-map construction, whose derivation is provided in Appendix C. As formalized in Appendix D, the non-zero singular values of the resulting composite Jacobian remain close to 1 for the small angular steps typical of low-rank fine-tuning, thereby preserving a more faithful and better-conditioned tangent-space backward geometry than Euclidean normalization.

Finally, because LoRA adapters are typically initialized so that the tangent update satisfies $U_j \to 0$ at the beginning of training, Geo-Flow must also handle the corresponding numerical singularity in a stable manner. In practice, this case is addressed through a numerically stable implementation that preserves continuity of both the forward computation and the backward gradient flow near the origin.

3.4 The Overall GO-LoRA Architecture

Integrating Ortho-Proj and Geo-Flow, the overall directional–magnitude recombination in GO-LoRA for the j-th column is given by

$$W_{:,j} = m_j \cdot \frac{v_j^{\text{new}}}{\|v_j^{\text{new}}\|_2} = m_j \cdot \frac{v_j^{\text{new}}}{\|w_j^0\|_2}. \tag{8}$$

Because $\|v_j^{\text{new}}\|_2 = \|w_j^0\|_2$ by exact norm preservation (Appendix B), the normalization factor remains fixed throughout training. This yields a clean decoupling: the magnitude adapter controls column-wise scaling, while the low-rank factors determine the tangent-space directional update through Ortho-Proj and Geo-Flow.

GO-LoRA preserves the lightweight trainable-parameter footprint characteristic of PEFT methods while introducing only modest additional computation during training through tangent projection and trigonometric evaluation. Detailed complexity analysis is provided in Appendix E, and complete pseudocode is given in Algorithm 1 and Appendix F.

Finally, although GO-LoRA employs non-linear Riemannian transformations during training, it supports an exact closed-form offline reparameterization after optimization. Once training is complete, the learned tangent-space trajectory can be folded into a static dense weight matrix $\hat{W}$, so inference requires no additional adaptation branch beyond the merged model.

4 Experiments

To evaluate the efficacy of GO-LoRA, we conduct extensive experiments spanning language understanding, mathematical and code generation, image classification, and multimodal vision-language reasoning. We compare GO-LoRA against LoRA, weight-decomposed DoRA [50], along with recent competitive variants including PiSSA [53], OLoRA [11], rsLoRA [39], TriLoRA [27], MoSLoRA [70], ScaledAdamW [77], StelLA [48], HiRA [35] and FVAE-LoRA [42] where applicable. Our fundamental question is whether intrinsically resolving capacity redundancy and Euclidean distortion translates into empirical performance gains without increasing the parameter budget. Hyperparameter details are given in Appendix G, and statistical significance tests are reported in Appendix I.

4.1 Natural Language Understanding: Commonsense Reasoning

Datasets. We evaluate models on eight standard commonsense reasoning datasets: BoolQ [20], PIQA [7], SIQA [62], HellaSwag [76], WinoGrande [61], ARC-e and ARC-c [21], and OBQA [54].

Implementation Details. Following the evaluation protocol of DoRA, we utilize LLaMA2-7B [66] and LLaMA3-8B [28] as our backbone models. All experiments are conducted on NVIDIA A100 80GB GPUs. Accuracy is chosen as the principal metric (Appendix G.1).

Results. As shown in Table 1, GO-LoRA consistently outperforms the other PEFT methods across both backbone models. The gains are especially clear on HellaSwag, while GO-LoRA also remains highly competitive on the harder ARC variants. On LLaMA2-7B, GO-LoRA improves the average accuracy from 82.35% (StelLA) to 82.67%, while on LLaMA3-8B it further increases the average score from 86.73% (StelLA) to 87.05%. This pattern is consistent with our theoretical picture: more

faithful manifold-aware directional updates can reduce post-hoc normalization distortion and improve the efficiency of low-rank adaptation during training, which may be especially helpful for downstream reasoning tasks [69]. Additional training-efficiency results are reported in Appendix H.

Table 1: Accuracy comparison of LLaMA2-7B and LLaMA3-8B with various PEFT methods on eight commonsense reasoning datasets. **Bold** indicates the best result and underlined indicates the second best. All results are averaged over 3 runs.

	METHOD	Params (%)	BoolQ	PIQA	SIQA	HellaS.	WinoG.	ARC-e	ARC-c	OBQA	Avg.
LLaMA2-7B	LoRA	0.83	72.02	83.51	79.87	90.44	82.64	84.86	71.23	81.50	80.76
	DoRA	0.84	72.61	83.49	79.88	90.83	83.54	85.20	71.29	81.17	81.00
	PiSSA	0.83	71.20	83.91	79.16	91.04	82.89	85.13	69.45	83.94	80.84
	OLoRA	0.83	71.05	82.72	78.58	89.37	81.43	83.59	68.12	80.23	79.39
	TriLoRA	0.83	71.22	80.99	78.37	80.97	77.64	81.81	66.66	79.80	77.18
	MoSLoRA	0.83	71.60	83.87	79.64	90.47	83.23	84.35	69.91	80.49	80.45
	ScaledAdamW	0.83	72.27	83.88	79.65	90.83	82.46	85.54	70.65	81.89	80.90
	StelLA	0.83	73.61	84.95	80.61	91.46	84.58	86.39	72.82	84.37	82.35
	HiRA	0.83	71.26	83.36	79.48	88.15	83.93	86.77	**73.82**	84.63	81.43
	GO-LoRA (Ours)	0.84	**73.90**	**85.20**	**81.00**	**91.80**	**84.80**	**86.80**	73.20	**84.66**	**82.67**
LLaMA3-8B	LoRA	0.70	75.22	88.17	80.13	95.42	86.69	90.89	78.66	86.95	85.27
	DoRA	0.71	75.35	88.01	79.97	95.29	86.27	90.50	79.74	86.13	85.16
	PiSSA	0.70	74.68	88.19	80.51	94.97	85.27	90.18	78.84	85.63	84.78
	OLoRA	0.70	74.43	87.72	79.52	94.73	85.37	90.05	78.21	84.95	84.37
	TriLoRA	0.70	73.06	86.64	78.66	93.37	82.90	87.78	75.30	84.38	82.76
	MoSLoRA	0.70	74.82	88.39	80.33	95.53	86.24	89.98	79.80	85.84	85.12
	ScaledAdamW	0.70	75.30	88.60	80.18	95.89	85.08	91.10	80.57	86.63	85.42
	StelLA	0.70	75.92	89.89	81.68	96.42	87.80	92.02	82.34	87.75	86.73
	HiRA	0.70	75.45	89.76	81.10	95.34	87.67	**93.28**	**82.92**	**88.27**	86.72
	GO-LoRA (Ours)	0.71	**76.20**	**90.20**	**82.00**	**96.75**	**88.15**	92.35	82.65	88.10	**87.05**

4.2 Math and Code Generation

Datasets. To assess GO-LoRA's capabilities, we evaluate on two axes: (1) mathematical reasoning, fine-tuning on MetaMathQA [74] and tested on GSM8K [22] and MATH [31]; and (2) code generation, training on CodeFeedback [80] and assessed on HumanEval [13] and MBPP [4].

Implementation Details. We adopt LLaMA2-7B as the backbone and compare GO-LoRA with LoRA [33], DoRA [50], PiSSA [53], and StelLA [48]. Low-rank adapters are injected into every linear layer across both attention and FFN blocks with rank $r = 128$ and scaling factor $\alpha = 128$. All methods are trained with the AdamW optimizer for 1 epoch using a batch size of 128 (Appendix G.2).

Results. As shown in Table 2, GO-LoRA achieves the highest average score of 39.83%, outperforming StelLA and LoRA by 0.54 and 2.06 points, respectively. The advantage is especially pronounced on multi-step math benchmarks: GO-LoRA reaches 65.85% on GSM8K and 17.10% on MATH, directly surpassing the strongest baselines StelLA (65.40%) and PiSSA (16.61%). This pattern is consistent with the view that Geo-Flow helps preserve gradient fidelity over long reasoning chains by reducing the distortion induced by standard secant approximations. GO-LoRA likewise leads code generation with 34.15% on HumanEval and 42.20% on MBPP, suggesting that tangent-space directional updates can also benefit structured output synthesis.

Table 2: Math and code generation results on LLaMA2-7B. All results are averaged over 3 runs.

Model	Method	Params (%)	GSM8K	MATH	HumanEval	MBPP	Avg.
LLaMA2-7B	LoRA	4.53	64.71	15.29	30.12	40.95	37.77
	DoRA	4.55	64.82	15.36	26.96	39.31	36.61
	PiSSA	4.53	63.77	16.61	31.15	38.18	37.43
	StelLA	4.58	65.40	16.58	33.68	41.51	39.29
	GO-LoRA (Ours)	4.55	**65.85**	**17.10**	**34.15**	**42.20**	**39.83**

4.3 Image Classification

Datasets. To verify that GO-LoRA can effectively extend to image tasks, we evaluate its parameter-efficient adaptation on a Vision Transformer (ViT) backbone [26, 52] across a stringent testbed of six diverse and challenging image classification benchmarks. These datasets include DTD [19], EuroSAT

[30], GTSRB [64], RESISC45 [16], SUN397 [71], and SVHN [55], spanning a wide range of visual domains and real-world granularity levels.

Implementation Details. We employ the ViT-B/16 pretrained on ImageNet-21K as the shared backbone across all tasks. GO-LoRA is compared against full FT and several representative LoRA-family methods, including LoRA [33], PiSSA [53], rsLoRA [39], DoRA [50], FVAE-LoRA [42] and OLoRA [11]. All low-rank methods share the same adapter placement and rank configuration for a controlled comparison. The evaluation metric is top-1 accuracy (Appendix G.3).

Results. As shown in Table 3, GO-LoRA achieves the highest average accuracy (90.28%), surpassing all baselines, including full FT (89.37%). We attribute this to GO-LoRA devoting its limited parameter budget more efficiently to directional adaptation through Ortho-Proj and Geo-Flow. Its advantages on domain-specific datasets such as DTD and SUN397 further suggest that the proposed geometry-aware directional updates remain effective when downstream distributions deviate substantially from ImageNet [23] pre-training, supporting the method's applicability across diverse vision tasks.

Table 3: Image classification results on ViT-B/16. **Bold** indicates the best result, and underlined indicates the second best. All results are averaged over 3 runs.

Method	Params (%)	DTD	EuroSAT	GTSRB	RESISC45	SUN397	SVHN	Avg.
Full FT	-	78.18	**98.28**	**98.80**	**94.29**	69.37	**97.32**	89.37
LoRA	0.72	74.65	97.28	96.97	90.10	71.07	94.27	87.39
PiSSA	0.72	74.19	97.31	96.96	89.76	69.06	94.86	87.02
rsLoRA	0.72	72.24	97.42	96.58	87.99	67.74	93.82	85.97
DoRA	0.75	75.74	97.22	97.29	91.76	71.58	96.41	88.33
OLoRA	0.72	72.28	96.66	97.14	88.98	69.68	94.85	86.60
FVAE-LoRA	0.73	78.19	97.82	98.03	93.53	73.14	96.53	89.54
GO-LoRA (Ours)	0.75	**79.20**	98.10	98.50	94.20	**74.80**	96.90	**90.28**

4.4 Multimodal Vision-Language Understanding

Datasets. To evaluate GO-LoRA's capacity for cross-modal alignment and complex multimodal reasoning, we assess it on four primary and widely used image-text benchmarks: VQA [3], GQA [36], NLVR^2 [65], and COCO Caption [49].

Implementation Details. Multimodal reasoning intrinsically requires more representational capacity to align heterogeneous vector spaces [72, 73]. Thus, testing PEFT frameworks on vision-language tasks is a stringent evaluation of their parameter utilization efficiency. Following previous work, we adopt the VL-BART [17] architecture and fine-tune it globally across four primary image-text understanding benchmarks: VQA, GQA, NLVR^2, and COCO Caption. We evaluate against FT, LoRA, and DoRA under the same task protocol and adapter placement (Appendix G.4).

Results. Table 4 presents the multi-task evaluation results. Under the reported aggregate score, GO-LoRA achieves the best overall result among the compared methods, reaching an average score of 78.01 while using only 5.96% of the trainable parameters. Compared with full FT, GO-LoRA performs better on NLVR^2 and COCO Caption, while trailing slightly on VQA and GQA. This pattern is consistent with the view that reducing capacity redundancy in the directional branch can improve adaptation efficiency and regularization in high-complexity multimodal settings.

Table 4: The multi-task evaluation results on VQA, GQA, NLVR^2, and COCO Caption utilizing the VL-BART backbone framework. GO-LoRA achieves the best reported aggregate score among the compared methods. All results are averaged over 3 runs.

Method	Params (%)	VQA	GQA	NLVR^2	COCO Cap	Avg.
Full FT	100	**66.86**	**56.74**	73.68	112.02	77.33
LoRA	5.93	65.24	53.56	71.94	115.28	76.51
DoRA	5.96	65.84	54.66	73.14	115.88	77.38
GO-LoRA (Ours)	5.96	66.34	55.47	**73.83**	**116.40**	**78.01**

4.5 Ablation Study

Component Analysis. To isolate the contributions of our proposed mechanisms, we progressively upgrade a decoupled baseline on LLaMA2-7B commonsense reasoning. We evaluate three configurations: (1) the vanilla Euclidean baseline; (2) adding Ortho-Proj to purify the update direction; and (3) adding Geo-Flow for exact geodesic reparameterization (Full GO-LoRA). We do not apply Geo-Flow in isolation, because the Riemannian Exponential Map requires an orthogonal tangent vector ($\langle w_j^0, U_j \rangle = 0$). Without Ortho-Proj, unconstrained radial noise can disrupt the intended manifold trajectory and degrade the pretrained directional structure.

Results. Table 5 demonstrates a clear progression. Applying Ortho-Proj alone improves accuracy from 81.00% to 81.60%, indicating that removing radial redundancy improves the utilization of the low-rank budget. When Geo-Flow is subsequently integrated, performance reaches 82.67%, yielding a 1.67-point improvement over the baseline. This supports the view that once Ortho-Proj supplies a valid tangent vector, Geo-Flow further improves the directional update by replacing the secant-style approximation with a geodesic construction.

4.6 Effective Rank Analysis

Evaluation Metric. To validate our capacity redundancy hypothesis, we measure the effective rank of learned updates using the participation ratio [60]: $\text{EffRank}(M) = \frac{(\sum_i \sigma_i)^2}{\sum_i \sigma_i^2}$, where σ_i are singular values of the update matrix BA. This captures spectral energy dispersion.

Results. Table 6 provides an empirical diagnostic of how the learned update energy is distributed across the available low-rank dimensions. Although all methods use the same nominal rank $r = 32$, LoRA and DoRA exhibit more concentrated singular-value spectra, resulting in lower participation-ratio effective ranks of 17.6 and 20.4, respectively. In contrast, GO-LoRA achieves a higher effective rank of 25.8, corresponding to 80.6% rank utilization, while also attaining the best average accuracy of 82.67%. This observation is consistent with our intuition that removing radial redundancy through tangent-space projection encourages a more balanced use of the available low-rank subspace.

Table 5: Ablation study on LLaMA2-7B commonsense reasoning (8 datasets average).

Configuration	Ortho-Proj	Geo-Flow	Avg.
Baseline	✗	✗	81.00
+ Ortho-Proj	✓	✗	81.60
+ Geo-Flow	✓	✓	**82.67**

Table 6: Effective rank analysis of learned updates (nominal rank $r = 32$) on LLaMA2-7B.

Method	Effective Rank	rate (%)	Avg.
LoRA	17.6	55.0	80.76
DoRA	20.4	63.8	81.00
GO-LoRA	**25.8**	**80.6**	**82.67**

4.7 Geometric Diagnostics and Parameter Efficiency

Radial Redundancy in Real Training. To examine the practical significance of radial redundancy, we analyze the learned low-rank residuals of DoRA during LLaMA2-7B commonsense fine-tuning with AdamW. The radial-energy fraction decreases from 30.2% at 10% training progress to 23.2% at convergence. Despite progressive adaptation, a substantial portion of the residual energy remains aligned with the pretrained direction. This observation supports the geometric motivation of GO-LoRA, where Ortho-Proj explicitly removes the redundant parallel component and directs the effective update into the tangent space.

Parameter-Matched Comparisons. We further evaluate GO-LoRA against vanilla LoRA under closely matched parameter budgets. For each GO-LoRA configuration, we select a slightly higher LoRA rank, ensuring that LoRA has more trainable parameters. As shown in Table 7, GO-LoRA consistently achieves higher accuracy across all evaluated rank settings.

Notably, at the main $r = 32$ setting, GO-LoRA achieves 82.67% accuracy using 56.98M trainable parameters, outperforming LoRA at $r = 33$ (80.80%) despite its larger 57.85M parameter budget. These results demonstrate that GO-LoRA improves adaptation efficiency through its geometric reparameterization rather than simply increasing trainable capacity.

Table 7: Parameter-matched comparisons on LLaMA2-7B commonsense reasoning. LoRA uses slightly more trainable parameters in each comparison.

Rank (GO/LoRA)	Params (M) (GO/LoRA)	LoRA	GO-LoRA	Δ
4/5	7.89/8.77	78.05	**79.50**	+1.45
8/9	14.90/15.78	79.05	**80.71**	+1.66
16/17	28.93/29.80	79.89	**81.82**	+1.93
32/33	56.98/57.85	80.80	**82.67**	+1.87

4.8 Rank Sensitivity Analysis

We examine how GO-LoRA behaves under varying rank budgets ($r \in \{4, 8, 16, 32, 64\}$) on LLaMA2-7B commonsense reasoning. As shown in Figure 2, GO-LoRA consistently outperforms all baselines at every rank, with the performance gap widening as the budget decreases. At $r = 4$, GO-LoRA achieves 79.50%, a $+1.8\%$ margin over the baseline, confirming that eliminating capacity redundancy is most impactful when rank resources are scarce.

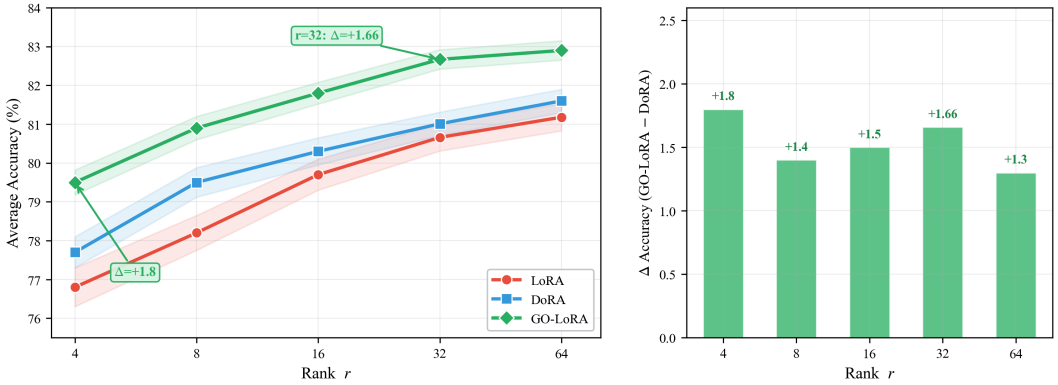


Figure 2: Rank sensitivity on LLaMA2-7B commonsense reasoning. Left: accuracy vs. rank for all methods. Right: accuracy gap (Δ) between GO-LoRA and DoRA narrows as rank increases, confirming that capacity redundancy elimination is most impactful under tight rank budgets.

5 Conclusion

In this paper, we address a geometric limitation in parameter-decoupled PEFT frameworks. By projecting directional updates onto the orthogonal tangent space (Ortho-Proj) and constructing geodesic-consistent directional parameterization via the Exponential Map (Geo-Flow), we propose the GO-LoRA framework. GO-LoRA mitigates the capacity redundancy and directional gradient distortion induced by heuristic L_2 normalization in decoupled directional branches. Through extensive experiments across language understanding, visual classification, multimodal reasoning, and mathematical and code generation tasks, we show that GO-LoRA consistently outperforms strong baselines. Our ablation studies confirm the complementary roles of Ortho-Proj and Geo-Flow, while effective-rank analysis supports the claim that GO-LoRA uses low-rank capacity more efficiently. Importantly, GO-LoRA delivers these gains across diverse architectures without additional inference-time overhead after offline merging. Looking forward, we hope this work highlights the value of bringing manifold-aware parameterization into PEFT and provides a useful foundation for future geometry-aware fine-tuning methods.

Acknowledgments and Disclosure of Funding

This work was supported in part by the New Generation Artificial Intelligence-National Science and Technology Major Project No. 2025ZD0124202, and the National Natural Science Foundation of China under Grants 62527806, 62125305, and U23A20339. And the authors declare no competing interests. We sincerely thank the reviewers for their valuable feedback and constructive suggestions.

References

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

[2] 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)*, pages 7319–7328, 2021.

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[4] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[5] Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? *Advances in Neural Information Processing Systems*, 31, 2018.

[6] Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *International Conference on Learning Representations (ICLR 2019)*, volume 9, pages 6384–6399, 2019.

[7] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.

[8] Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[9] Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

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

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

[12] Yuanyang Cao, Xichun Liu, Fuwei Zhang, Shangqi Deng, Ziyang Ren, and Jianji Wang. Beyond static allocation: Dynamic sensitivity-aware fine-tuning for vision transformers. In *Forty-third International Conference on Machine Learning*, 2026.

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

[14] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

[15] Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[16] 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.

[17] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR, 2021.

[18] Minhyung Cho and Jaehyung Lee. Riemannian approach to batch normalization. *Advances in Neural Information Processing Systems*, 30, 2017.

[19] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[20] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pages 2924–2936, 2019.

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

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

[23] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[24] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

[25] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235, 2023.

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

[27] Chengcheng Feng, Mu He, Qiuyu Tian, Haojie Yin, Xiaofang Zhao, Hongwei Tang, and Xingqiang Wei. Trilora: Integrating svd for advanced style personalization in text-to-image generation. *arXiv preprint arXiv:2405.11236*, 2024.

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

[29] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+ efficient low rank adaptation of large models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 17783–17806, 2024.

[30] 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.

[31] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[32] 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*, pages 2790–2799. PMLR, 2019.

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

[34] Lei Huang, Xianglong Liu, Bo Lang, Adams Yu, Yongliang Wang, and Bo Li. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[35] Qiushi Huang, Tom Ko, Zhan Zhuang, Lilian Tang, and Yu Zhang. Hira: Parameter-efficient hadamard high-rank adaptation for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[36] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[37] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pages 709–727. Springer, 2022.

[38] Haodong Jing, Yongqiang Ma, Wenjie Gao, Dongyao Jiang, Shuai Huang, and Nanning Zheng. Mind-vad: Brain-inspired fmri-to-video precise reconstruction via cross-modal autoregressive diffusion. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026.

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

[40] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[41] Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. Vera: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations*, 2024.

[42] Shashi Kumar, Yacouba Kaloga, John Mitros, Petr Motlicek, and Ina Kodrasi. Latent space factorization in LoRA. *arXiv preprint arXiv:2510.19640*, 2025.

[43] John M Lee. *Introduction to Riemannian manifolds*, volume 2. Springer, 2018.

[44] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 3045–3059, 2021.

[45] Jun Li, Fuxin Li, and Sinisa Todorovic. Efficient riemannian optimization on the stiefel manifold via the cayley transform. In *International Conference on Learning Representations*, 2020.

[46] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[47] Xiang Lisa Li and Percy 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)*, pages 4582–4597, 2021.

[48] Zhizhong Li, Sina Sajadmanesh, Jingtao Li, and Lingjuan Lyu. Stella: Subspace learning in low-rank adaptation using stiefel manifold. *arXiv preprint arXiv:2510.01938*, 2025.

[49] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[50] 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.

[51] Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, et al. Parameter-efficient orthogonal finetuning via butterfly factorization. In *The Twelfth International Conference on Learning Representations*, 2024.

[52] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

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

[54] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2381–2391, 2018.

[55] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4. Granada, 2011.

[56] Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *Advances in neural information processing systems*, 30, 2017.

[57] Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. *Advances in Neural Information Processing Systems*, 36:79320–79362, 2023.

[58] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[59] Adithya Renduchintala, Tugrul Konuk, and Oleksii Kuchaiev. Tied-lora: Enhancing parameter efficiency of lora with weight tying. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8694–8705, 2024.

[60] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pages 606–610. IEEE, 2007.

[61] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[62] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 4463–4473, 2019.

[63] A Saxe, J McClelland, and S Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *Proceedings of the International Conference on Learning Representations*. ICLR, 2014.

[64] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011.

[65] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 6418–6428, 2019.

[66] 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.

[67] 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*, pages 3274–3287, 2023.

[68] Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Lora-pro: Are low-rank adapters properly optimized? *arXiv preprint arXiv:2407.18242*, 2024.

[69] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[70] Taiqiang Wu, Jiahao Wang, Zhe Zhao, and Ngai Wong. Mixture-of-subspaces in low-rank adaptation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7880–7899, 2024.

[71] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[72] Panqi Yang, Haodong Jing, Nanning Zheng, and Yongqiang Ma. Unihoi: Unified human-object interaction understanding via unified token space. *arXiv preprint arXiv:2511.15046*, 2025.

[73] Panqi Yang, Haodong Jing, Nanning Zheng, and Yongqiang Ma. Instrucrobo: Object-centric multi-instruction decoupling model for explainable robotic manipulation. *Engineering Applications of Artificial Intelligence*, 2026.

[74] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

[75] 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)*, pages 1–9, 2022.

[76] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800, 2019.

[77] Fangzhao Zhang and Mert Pilanci. Riemannian preconditioned lora for fine-tuning foundation models. *arXiv preprint arXiv:2402.02347*, 2024.

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

[79] Xiao Zhang, Haodong Jing, Yongqiang Ma, and Nanning Zheng. Decoupling primitive with experts: Dynamic feature alignment for compositional zero-shot learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

[80] Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. Opencodeinterpreter: Integrating code generation with execution and refinement. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12834–12859, 2024.

A Proof of Capacity Redundancy (Proposition 1)

Here we formalize the notion of capacity redundancy introduced in the main text by showing that, under the normalized directional update mapping used in decoupled PEFT, the component parallel to the base direction contributes no independent directional degree of freedom.

Proposition 1. Let $v_j \neq 0$ denote the base direction in a decoupled directional branch. Let $\delta_j \in \mathbb{R}^d$ be an unconstrained low-rank update (e.g., $\delta_j = (BA)_{:,j}$). We decompose δ_j into two mutually orthogonal components: the parallel component $\delta_j^{\parallel} = \mathrm{proj}_{v_j}(\delta_j) \in \mathrm{span}(v_j)$ and the orthogonal component $\delta_j^{\perp} = \delta_j - \delta_j^{\parallel} \in T_{v_j}\mathcal{M}$. Let $f(v_j, \delta_j) = \frac{v_j+\delta_j}{\|v_j+\delta_j\|_2}$ be the standard normalized update mapping. Then for any $\alpha > -1$,

$$f(v_j,\ \alpha v_j + \delta_j^{\perp}) \ =\ f\Big(v_j,\ \tfrac{1}{1+\alpha}\,\delta_j^{\perp}\Big)\,, \tag{9}$$

i.e., under the normalized update mapping, the parallel component $\delta_j^{\parallel} = \alpha v_j$ contributes no independent directional information; its sole effect is to rescale the effective orthogonal perturbation, which can equivalently be achieved by adjusting $\|\delta_j^{\perp}\|_2$ alone.

Proof. Let $\delta_j^{\parallel} = \alpha v_j$ for some scalar $\alpha \in \mathbb{R}$. Substituting this decomposition into the mapping function gives

$$\begin{aligned} f(v_j, \delta_j) &= \frac{v_j + (\alpha v_j + \delta_j^{\perp})}{\|v_j + \alpha v_j + \delta_j^{\perp}\|_2} \\ &= \frac{(1+\alpha)v_j + \delta_j^{\perp}}{\sqrt{\|(1+\alpha)v_j\|_2^2 + \|\delta_j^{\perp}\|_2^2}} \\ &= \frac{v_j + \frac{1}{1+\alpha}\delta_j^{\perp}}{\sqrt{\|v_j\|_2^2 + \frac{1}{(1+\alpha)^2}\|\delta_j^{\perp}\|_2^2}}, \end{aligned} \tag{10}$$

where the second equality uses the orthogonality condition $\langle v_j, \delta_j^{\perp}\rangle = 0$, and the third equality follows from dividing the numerator and denominator by $(1+\alpha)$, which is valid since $\alpha > -1$.

The final expression is functionally identical to $f(v_j, \frac{1}{1+\alpha}\delta_j^{\perp})$. Hence, the parallel component αv_j introduces no independent directional degree of freedom after normalization; it merely rescales the effective orthogonal step by a factor of $\frac{1}{1+\alpha}$. Therefore, any portion of the update devoted to the component parallel to v_j can be equivalently absorbed into a rescaling of the orthogonal component, making it redundant in the normalized directional branch. □

B Proof of Norm Preservation (Theorem 2)

In Section 3, we stated that Geo-Flow preserves the norm of the reparameterized directional vector and therefore keeps it on the same sphere anchored at the pretrained weight. We now provide the formal proof.

Theorem 2. Let $w_j^0 \in \mathcal{S}^{d-1}(R)$ with $R = \|w_j^0\|_2$, where w_j^0 denotes the frozen pretrained anchor used by GO-LoRA. Assume a non-zero projected tangent update $U_j \in T_{w_j^0}\mathcal{M}_j$ (meaning $\langle w_j^0, U_j\rangle = 0$ and $U_j \neq 0$). The reparameterized directional vector is defined via the exponential map as:

$$v_j^{new} = w_j^0 \cos(\theta_j) + \frac{U_j}{\|U_j\|_2}\|w_j^0\|_2 \sin(\theta_j), \tag{11}$$

where $\theta_j = \frac{\|U_j\|_2}{\|w_j^0\|_2}$. (For the trivial case $U_j = 0$, the mapping degenerates to the identity $v_j^{new} = w_j^0$). This mapping is exactly norm-preserving: $\|v_j^{new}\|_2 \equiv \|w_j^0\|_2$.

Proof. We compute the squared L_2 norm of v_j^{new}:

$$\|v_j^{new}\|_2^2 = \left\langle w_j^0 \cos(\theta_j) + \frac{U_j}{\|U_j\|_2}\|w_j^0\|_2 \sin(\theta_j),\ w_j^0 \cos(\theta_j) + \frac{U_j}{\|U_j\|_2}\|w_j^0\|_2 \sin(\theta_j) \right\rangle. \tag{12}$$

Expanding the inner product into three terms (self-a, cross, self-b) yields:

$$\|v_j^{new}\|_2^2 = \underbrace{\|w_j^0\|_2^2 \cos^2(\theta_j)}_{\|a\|^2} + \underbrace{2\cos(\theta_j)\sin(\theta_j)\frac{\|w_j^0\|_2}{\|U_j\|_2}\langle w_j^0, U_j\rangle}_{2\langle a,b\rangle} + \underbrace{\|w_j^0\|_2^2 \sin^2(\theta_j)}_{\|b\|^2}, \tag{13}$$

where $a = w_j^0 \cos\theta_j$ and $b = \frac{U_j}{\|U_j\|_2}\|w_j^0\|_2 \sin(\theta_j)$. Because $U_j \in T_{w_j^0}\mathcal{M}_j$, the tangent-space orthogonality condition guarantees $\langle w_j^0, U_j\rangle = 0$, eliminating the cross term. This simplifies to:

$$\|v_j^{new}\|_2^2 = \|w_j^0\|_2^2 \cos^2(\theta_j) + \|w_j^0\|_2^2 \sin^2(\theta_j) = \|w_j^0\|_2^2(\cos^2(\theta_j) + \sin^2(\theta_j)) = \|w_j^0\|_2^2. \tag{14}$$

Taking the square root proves that $\|v_j^{new}\|_2 = \|w_j^0\|_2 = R$. □

C Gradient Derivation and Backpropagation of Geo-Flow

We derive the analytical gradients for backpropagating through the Riemannian Exponential Map and characterize the corresponding Jacobian under GO-LoRA's frozen-anchor parameterization. Let the loss function be $\mathcal{L}$. During backpropagation, we receive the upstream gradient $\frac{\partial \mathcal{L}}{\partial v_j^{new}}$ from the downstream computation. The following derivation assumes $U_j \neq 0$; the singular case $U_j = 0$ is handled separately in implementation.

By taking the derivative of the Geo-Flow forward pass (Eq. 7) with respect to the orthogonal update vector U_j, we obtain the Jacobian matrix $\mathcal{J} = \frac{\partial v_j^{new}}{\partial U_j} \in \mathbb{R}^{d\times d}$.

Let $\hat{U}_j = \frac{U_j}{\|U_j\|_2}$ and $\hat{w}_j^0 = \frac{w_j^0}{\|w_j^0\|_2}$. Differentiating $\theta_j = \frac{\|U_j\|_2}{\|w_j^0\|_2}$ with respect to U_j yields $\nabla_{U_j}\theta_j = \frac{\hat{U}_j}{\|w_j^0\|_2}$. The Jacobian is derived as

$$\begin{aligned}\frac{\partial v_j^{new}}{\partial U_j} &= \frac{\partial}{\partial U_j}\left(w_j^0 \cos(\theta_j)\right) + \frac{\partial}{\partial U_j}\left(\hat{U}_j \|w_j^0\|_2 \sin(\theta_j)\right) \\ &= -w_j^0 \sin(\theta_j)\left(\frac{\hat{U}_j}{\|w_j^0\|_2}\right)^T + \|w_j^0\|_2\left(\frac{\partial \hat{U}_j}{\partial U_j}\sin(\theta_j) + \hat{U}_j\cos(\theta_j)\left(\frac{\hat{U}_j}{\|w_j^0\|_2}\right)^T\right).\end{aligned} \tag{15}$$

Applying

$$\frac{\partial \hat{U}_j}{\partial U_j} = \frac{1}{\|U_j\|_2}\left(I - \hat{U}_j\hat{U}_j^T\right),$$

the full Jacobian condenses into

$$\mathcal{J} = \underbrace{\frac{\|w_j^0\|_2}{\|U_j\|_2}\sin(\theta_j)\left(I - \hat{U}_j\hat{U}_j^T\right)}_{\text{Orthogonal mapping flow}} + \underbrace{\left(\hat{U}_j\cos(\theta_j) - \hat{w}_j^0\sin(\theta_j)\right)}_{\text{Curvature correction}}\hat{U}_j^T. \tag{16}$$

This Jacobian governs gradient propagation under the manifold-constrained update. The gradient with respect to the tangent update is

$$\nabla_{U_j}\mathcal{L} = \mathcal{J}^T\nabla_{v_j^{new}}\mathcal{L}. \tag{17}$$

Here $P_{w_j^0} = I - \hat{w}_j^0(\hat{w}_j^0)^T$ denotes the orthogonal projector onto the tangent space at the frozen anchor w_j^0. Since $U_j = P_{w_j^0}\delta_j$, the gradient with respect to the pre-projection update δ_j is

$$\nabla_{\delta_j}\mathcal{L} = P_{w_j^0}^T\mathcal{J}^T\nabla_{v_j^{new}}\mathcal{L} = P_{w_j^0}\mathcal{J}^T\nabla_{v_j^{new}}\mathcal{L}, \tag{18}$$

which is the quantity that further backpropagates to the low-rank factors A and B.

D Comparison of Backward Geometry Under Normalized and Geodesic Directional Updates (Theorem 3)

Standard decoupled PEFT normalizes linear addition as $f(\delta_j) = \frac{v_j+\delta_j}{\|v_j+\delta_j\|_2}$. Below we compare the Jacobian geometry of this heuristic normalization with that of GO-LoRA's tangent-projected geodesic construction anchored at the pretrained weight.

Theorem 3. Let $f(\delta_j) = \frac{v_j+\delta_j}{\|v_j+\delta_j\|_2}$ denote the Euclidean normalization operator, and let $g(\delta_j) = \mathrm{Exp}_{w_j^0}(P_{w_j^0}\delta_j)$ denote the GO-LoRA directional construction, where $P_{w_j^0} = I - \hat{w}_j^0(\hat{w}_j^0)^T$, $\hat{w}_j^0 = \frac{w_j^0}{\|w_j^0\|_2}$, and $\theta_j = \|P_{w_j^0}\delta_j\|_2/\|w_j^0\|_2$.

(i) The Jacobian of f is

$$\left.\frac{\partial f}{\partial \delta}\right|_{x=v_j+\delta_j} = \frac{1}{\|v_j+\delta_j\|_2}\left(I - \hat{x}_j\hat{x}_j^T\right), \qquad \hat{x}_j = \frac{v_j+\delta_j}{\|v_j+\delta_j\|_2}. \tag{19}$$

Hence its singular values are 0 and $\frac{1}{\|v_j+\delta_j\|_2}$, so

$$\left\|\frac{\partial f}{\partial \delta}\right\|_2 = \frac{1}{\|v_j+\delta_j\|_2}.$$

Under the local fine-tuning regime $\|\delta_j\|_2 \ll \|v_j\|_2$, this scales as $\mathcal{O}(1/\|v_j\|_2)$ for the normalized directional branch itself.

(ii) Let $\mathcal{J} = \frac{\partial \mathrm{Exp}_{w_j^0}(U_j)}{\partial U_j}$ denote the Jacobian of the exponential map with respect to the tangent vector $U_j = P_{w_j^0}\delta_j$. Then the composite Jacobian of GO-LoRA with respect to δ_j is

$$Dg_{\delta_j} = \mathcal{J}P_{w_j^0}. \tag{20}$$

Its non-zero singular values are exactly 1 and $\frac{\sin\theta_j}{\theta_j}$, with one additional zero singular value along the radial direction annihilated by $P_{w_j^0}$. Hence, for small angular displacements θ_j, all non-zero singular values remain close to 1, preserving a more faithful and better-conditioned tangent-space Jacobian structure than post-hoc Euclidean normalization.

Proof of (i). Standard differentiation of vector normalization gives

$$\left.\frac{\partial f}{\partial \delta}\right|_{x=v_j+\delta_j} = \frac{1}{\|v_j+\delta_j\|_2}\left(I - \hat{x}_j\hat{x}_j^T\right), \qquad \hat{x}_j = \frac{v_j+\delta_j}{\|v_j+\delta_j\|_2}. \tag{21}$$

The matrix $(I - \hat{x}_j\hat{x}_j^T)$ is an orthogonal projector with eigenvalues $\{0, 1, \ldots, 1\}$. Hence the Jacobian has singular values 0 along $\hat{x}_j$ and $\frac{1}{\|v_j+\delta_j\|_2}$ on the orthogonal complement, implying

$$\left\|\frac{\partial f}{\partial \delta}\right\|_2 = \frac{1}{\|v_j+\delta_j\|_2}.$$

When $\|\delta_j\|_2 \ll \|v_j\|_2$, we have $\|v_j+\delta_j\|_2 = \Theta(\|v_j\|_2)$, and therefore the branch-level scaling is $\mathcal{O}(1/\|v_j\|_2)$.

Proof of (ii). From Appendix C, the Jacobian $\mathcal{J}$ acts on the tangent vector $U_j = P_{w_j^0}\delta_j$. Because $P_{w_j^0}$ annihilates the radial direction, it suffices to analyze the action on the tangent space $T_{w_j^0}\mathcal{M}_j$. Let $\hat{U}_j = \frac{U_j}{\|U_j\|_2}$. Then

$$\begin{aligned} \mathcal{J}h &= \frac{\sin\theta_j}{\theta_j}h, && h \perp \hat{U}_j,\ \hat{w}_j^0, \\ \mathcal{J}\hat{U}_j &= \hat{U}_j\cos\theta_j - \hat{w}_j^0\sin\theta_j, && \|\mathcal{J}\hat{U}_j\|_2 = 1. \end{aligned} \tag{22}$$

Thus, on the tangent subspace, the singular values are exactly 1 and $\frac{\sin\theta_j}{\theta_j}$. Composing with $P_{w_j^0}$ adds a zero singular value along the radial direction and leaves the non-zero singular values unchanged.

Therefore, the non-zero singular values of $Dg_{\delta_j} = \mathcal{J}P_{w_j^0}$ are exactly 1 and $\frac{\sin\theta_j}{\theta_j}$. By the Maclaurin expansion

$$\frac{\sin\theta}{\theta} = 1 - \frac{\theta^2}{6} + \cdots,$$

both non-zero singular values remain close to unity for small angular displacements. Hence GO-LoRA preserves a more faithful tangent-space backward geometry and avoids post-hoc radial waste by removing radial components before the geodesic update. □

E Computational Complexity Analysis

In this section, we analyze the computational and memory footprint of GO-LoRA. Consider a single linear layer mapping from dimension d to dimension k, with low-rank adapters of rank r. Let $X \in \mathbb{R}^{B\times L\times d}$ denote the input activation tensor, where B is the batch size and L is the sequence length.

Training Complexity. Similar to weight-decomposed methods such as DoRA [50], applying column-wise geometric transformations requires materializing the adapted weight matrix $W^{new} \in \mathbb{R}^{d\times k}$ during training. The forward pass consists of three main steps:

1. **Low-rank expansion:** Forming the unconstrained update $\Delta W = BA$ costs $\mathcal{O}(dkr)$ FLOPs.
2. **Geometric transformations:** Given ΔW, applying Ortho-Proj and Geo-Flow column-wise requires $\mathcal{O}(dk)$ additional operations for inner products, norms, and trigonometric combinations.
3. **Linear projection:** The forward pass then applies a dense projection XW^{new}, which costs $\mathcal{O}(BLdk)$ FLOPs. This has the same asymptotic order as the dense base projection already present in standard LoRA and the underlying backbone model.

Although materializing W^{new} changes the operation order relative to standard LoRA (which computes $XW_0 + (XB)A$), the dominant cost in modern LLM fine-tuning remains the dense projection and the associated activation memory. Because GO-LoRA optimizes only the low-rank factors and the magnitude vector, its trainable-parameter footprint remains firmly in the PEFT regime. While materializing W^{new} may introduce a temporary dense buffer during training, the primary optimizer-state and trainable-parameter savings relative to full fine-tuning are preserved.

Inference Complexity and Exact Reparameterization. While GO-LoRA relies on non-linear Riemannian transformations during training, these operations depend only on parameters that are fixed after training: the frozen base weight W_0, the learned low-rank matrices A and B, and the magnitude vector m. Therefore, GO-LoRA admits an exact offline reparameterization into a static dense weight matrix.

Let $\Delta W = BA$ be the learned low-rank update. For each column j, we compute the orthogonal tangent vector with respect to the pretrained anchor $w_j^0 = W_{0[:,j]}$

$$U_j = \Delta W_{:j} - \mathrm{proj}_{W_{0,j}}(\Delta W_{:j}),$$

and the angular displacement

$$\theta_j = \|U_j\|_2 / \|W_{0,j}\|_2.$$

The merged dense weight column is then

$$\hat{W}_{:j} = m_j \left(\frac{W_{0,j}}{\|W_{0,j}\|_2} \cos(\theta_j) + \frac{U_j}{\|U_j\|_2} \sin(\theta_j) \right). \tag{23}$$

For the trivial case $U_j = 0$, this reduces to $\hat{W}_{:j} = m_j \frac{W_{0,j}}{\|W_{0,j}\|_2}$.

This merge is performed entirely offline prior to deployment. Consequently, inference with the merged weight $\hat{W}$ uses the same dense linear operation $Y = X\hat{W}$ as the original backbone. Thus, GO-LoRA introduces no additional runtime adaptation branch at inference time and preserves the same asymptotic FLOP count as the base model, with no algorithmic latency overhead beyond the offline merge.

F GO-LoRA Algorithm Framework

Algorithm 1 provides the complete pseudocode for GO-LoRA's training procedure, including the Ortho-Proj and Geo-Flow steps.

Algorithm 1 GO-LoRA: Geodesic Reparameterization for Low-Rank Adaptation

Require: Pre-trained weight $W_0 \in \mathbb{R}^{d\times k}$, loss function $\mathcal{L}$, a Euclidean optimizer's step function `step`, rank r, scale factor α, number of iterations T, numerical stabilizer $\epsilon = 10^{-8}$.

Initialize LoRA factors $B_0 \in \mathbb{R}^{d\times r}$ (Kaiming), $A_0 \in \mathbb{R}^{r\times k}$ (zero).
Initialize magnitude $\mathbf{m}_0 \in \mathbb{R}^k$ where $m_{0,j} = \|W_0[:, j]\|_2$
Set scaling factor $\gamma = \alpha/r$

for $t \leftarrow 0$ **to** $T-1$ **do**
  ▷ *Compute low-rank update*
  $\Delta W_t \leftarrow \gamma \cdot B_t A_t$ ▷ $\Delta W_t \in \mathbb{R}^{d\times k}$

  ▷ *Ortho-Proj: project onto the tangent space at the anchor* w_j^0
  for $j = 1, \ldots, k$ **do**
    $w_j^0 \leftarrow W_0[:, j], \quad \delta_j \leftarrow \Delta W_t[:, j], \quad \hat{w}_j^0 \leftarrow w_j^0/\|w_j^0\|_2$
    $U_j \leftarrow \delta_j - \langle \delta_j, \hat{w}_j^0 \rangle \hat{w}_j^0$ ▷ Gram-Schmidt tangent proj
  end for

  ▷ *Geo-Flow: exponential map on the anchored sphere*
  for $j = 1, \ldots, k$ **do**
    ▷ *Numerically stable handling of the* $U_j \to 0$ *case*
    $\theta_j \leftarrow \|U_j\|_2/\|w_j^0\|_2, \quad \hat{U}_j \leftarrow U_j/(\|U_j\|_2 + \epsilon)$
    $v_j^{\text{new}} \leftarrow \|w_j^0\|_2 \cos(\theta_j)\hat{w}_j^0 + \operatorname{sinc}(\theta_j) U_j$ ▷ Geodesic construction
  end for

  ▷ *Magnitude–direction recombination*
  for $j = 1, \ldots, k$ **do**
    $W_t'[:, j] \leftarrow m_{t,j} \cdot v_j^{\text{new}}/\|w_j^0\|_2$ ▷ $\|v_j^{\text{new}}\|_2 = \|w_j^0\|_2$ by exact isometry
  end for

  ▷ *Forward pass and gradient computation*
  Compute loss: $\mathcal{L}_t \leftarrow \mathcal{L}(W_t'; \mathcal{B}_t)$. ▷ $\mathcal{B}_t$: mini-batch
  Compute gradients: $\nabla_{B_t}, \nabla_{A_t}, \nabla_{\mathbf{m}_t}$ ▷ via automatic diff

  ▷ *Euclidean optimizer step (e.g., AdamW)*
  $B_{t+1} \leftarrow \texttt{step}(B_t, \nabla_{B_t})$
  $A_{t+1} \leftarrow \texttt{step}(A_t, \nabla_{A_t})$
  $\mathbf{m}_{t+1} \leftarrow \texttt{step}(\mathbf{m}_t, \nabla_{\mathbf{m}_t})$
end for

▷ *Exact offline inference merge*
$\Delta W_T \leftarrow \gamma \cdot B_T A_T$
for $j = 1, \ldots, k$ **do**
  $w_j^0 \leftarrow W_0[:, j], \quad \delta_j \leftarrow \Delta W_T[:, j], \quad \hat{w}_j^0 \leftarrow w_j^0/\|w_j^0\|_2$
  $U_j \leftarrow \delta_j - \langle \delta_j, \hat{w}_j^0 \rangle \hat{w}_j^0, \quad \theta_j \leftarrow \|U_j\|_2/\|w_j^0\|_2, \quad \hat{U}_j \leftarrow U_j/(\|U_j\|_2 + \epsilon)$
  $\hat{W}[:, j] \leftarrow m_{T,j} \left(\cos(\theta_j)\hat{w}_j^0 + \frac{\operatorname{sinc}(\theta_j)}{\|w_j^0\|_2} U_j \right)$ ▷ Closed-form exact reparam.
end for
return Static deployed weight $\hat{W}$

G Experimental Configurations

This section provides complete hyperparameter configurations for all experiments reported in the main text. Unless otherwise noted, all PEFT methods (LoRA, DoRA, GO-LoRA, etc.) share the same general training hyperparameters within each experiment to ensure fair comparison. GO-LoRA additionally enables `orthogonal_direction` (Ortho-Proj) and `geodesic_direction` (Geo-Flow), with `geodesic_detach_trig=False` so that the exact Jacobian of the Exponential Map (derived in Appendix C) participates fully in backpropagation, realizing the tangent-space backward-geometry advantage described in Theorem 3. The scaling factor is computed as $\gamma = \alpha/r$ following the standard LoRA convention. LoRA B is initialized via Kaiming uniform and LoRA A via zero initialization; the decoupled magnitude vector $\mathbf{m}$ is initialized to the per-column norm of the pre-trained weight, i.e., $m_j = \|W_0[:, j]\|_2$.

System-Level Numerical Stability Design. A key challenge in GO-LoRA's geodesic reparameterization under mixed-precision training (e.g., FP16/BF16) is the numerical singularity at initialization. Since one low-rank factor is zero-initialized, the tangent update $U_j \to 0$ and the angular displacement $\theta_j \to 0$ at the start of training. To improve numerical stability and avoid NaN gradients or loss spikes, GO-LoRA uses the following implementation safeguards:

1. **Denominator Clamping:** All norm denominators (e.g., $\|w_j^0\|_2$) are lower-bounded using $\max(\cdot, \epsilon)$ with $\epsilon = 10^{-8}$ to avoid division by zero.
2. **Gradient-Safe Square Root:** When computing $\|U_j\|_2 = \sqrt{\sum_i U_{j,i}^2}$, we add a micro-perturbation $\epsilon_{\text{grad}} = 10^{-12}$ inside the square root to keep the backward derivative bounded near zero.
3. **Taylor Expansion at Singularity:** Evaluating Geo-Flow requires the sinc term $\sin(\theta_j)/\theta_j$. As $\theta_j \to 0$, this leads to a numerical singularity of the form $0/0$. When $\theta_j \leq 10^{-4}$, we switch to the 4th-order Taylor approximation $1 - \theta_j^2/6 + \theta_j^4/120$.
4. **Tangent Projection Safeguard:** The Gram-Schmidt orthogonalization uses the same denominator clamping strategy to preserve numerical stability in the tangent projection step.

These design choices ensure numerically stable mixed-precision training in our implementation and were used consistently across all reported experiments.

G.1 Commonsense Reasoning (LLaMA Family)

Table 8 summarizes the hyperparameters for commonsense reasoning experiments.

Table 8: Hyperparameters for commonsense reasoning experiments using LLaMA2-7B / LLaMA3-8B.

Parameter	Value / Setting
General Training Parameters	
Backbone Models	LLaMA2-7B, LLaMA3-8B
Hardware	NVIDIA A100 80GB GPUs
Optimizer	AdamW ($\beta_1 = 0.9,\ \beta_2 = 0.999$)
Learning Rate	2×10^{-4}
LR Scheduler	Cosine Annealing
Warmup Steps	200
Batch Size	16
Number of Epochs	2
Weight Decay	0.01
GO-LoRA Parameters	
Rank (r)	32
Alpha (α)	64
Dropout	0.05
Target Modules	`q_proj, k_proj, v_proj, up_proj, down_proj`

G.2 Math and Code Generation (LLaMA2-7B)

Table 9: Hyperparameters for math and code generation experiments using LLaMA2-7B.

Parameter	Value / Setting
General Training Parameters	
Backbone Model	LLaMA2-7B
Optimizer	AdamW ($\beta_1 = 0.9,\ \beta_2 = 0.999$)
Peak Learning Rate	2×10^{-4}
LR Scheduler	Cosine Annealing
Batch Size	128
Number of Epochs	1
Weight Decay	0.01
GO-LoRA Parameters	
Rank (r)	128
Alpha (α)	128
Target Modules	All linear layers (Attention + FFN)
Dataset Details	
Math Training Data	MetaMathQA [74] (395K)
Math Evaluation	GSM8K [22], MATH [31]
Code Training Data	CodeFeedback [80] (66K)
Code Evaluation	HumanEval [13], MBPP [4]

G.3 Image Classification (ViT-B/16)

Table 10 summarizes the hyperparameters for image classification experiments.

Table 10: Hyperparameters for image classification experiments using ViT-B/16 across six datasets.

Parameter	Value / Setting
General Training Parameters	
Backbone Model	ViT-B/16 (ImageNet-21K pre-trained)
Optimizer	AdamW ($\beta_1 = 0.9,\ \beta_2 = 0.999$)
Learning Rate	5×10^{-4}
LR Scheduler	Linear
Warmup Ratio	0.1
Batch Size	32
Number of Epochs	30
Weight Decay	0.01
GO-LoRA Parameters	
Rank (r)	16
Dropout	0.1
Target Modules	`query, key`
Datasets	
Benchmarks	DTD, EuroSAT, GTSRB, RESISC45, SUN397, SVHN
Evaluation Metric	Top-1 Accuracy

G.4 Multimodal Vision-Language Understanding (VL-BART)

Table 11 summarizes the hyperparameters for multimodal Vision-Language understanding experiments.

Table 11: Hyperparameters for multimodal Vision-Language understanding experiments using VL-BART.

Parameter	Value / Setting
General Training Parameters	
Backbone Model	VL-BART
Optimizer	AdamW ($\beta_1 = 0.9,\ \beta_2 = 0.999$)
Learning Rate	1×10^{-3}
LR Scheduler	Linear Decay
Warmup Ratio	0.1
Batch Size	300
Number of Epochs	20
Weight Decay	0.01
GO-LoRA Parameters	
Rank (r)	128
Alpha (α)	128
Dropout	0.05
Target Modules	`Q, K`
Datasets	
Benchmarks	VQA, GQA, NLVR^2, COCO Caption
VQA / GQA Metric	Accuracy
COCO Caption Metric	CIDEr

H Training Efficiency and Computational Overhead

We profile the peak GPU memory footprint during both training and evaluation to assess the practical overhead of GO-LoRA. Unless otherwise noted, the reported metrics are measured on NVIDIA A100 80GB GPUs.

Table 12 shows that on LLaMA2-7B commonsense reasoning, GO-LoRA increases training memory only slightly relative to DoRA (29.1 GB vs. 28.2 GB), while maintaining the same evaluation-time memory footprint of 13.5 GB. At the same time, GO-LoRA achieves a higher final accuracy of 82.67%, indicating that the geometric reparameterization improves performance with only modest additional training-time memory overhead.

Table 12: Training and evaluation memory comparison on LLaMA2-7B commonsense reasoning (NVIDIA A100 80GB).

Method	Train Memory (GB)	Eval Memory (GB)	Final Acc.
DoRA	28.2	13.5	81.00
GO-LoRA	**29.1**	**13.5**	**82.67**

Furthermore, Table 13 provides the corresponding memory comparison across model scales. In both LLaMA2-7B and LLaMA3-8B, GO-LoRA requires only about 0.9–1.0 GB more training memory than DoRA, while preserving the same evaluation-time memory footprint after merging and delivering consistently better accuracy.

Table 13: Detailed memory analysis across model scales (NVIDIA A100 80GB).

Model	Method	Train Memory (GB)	Eval Memory (GB)	Accuracy
LLaMA2-7B	DoRA	28.2	13.5	81.00
	GO-LoRA	29.1	13.5	**82.67**
LLaMA3-8B	DoRA	30.3	15.5	85.16
	GO-LoRA	31.3	15.5	**87.05**

I Statistical Significance Testing

Given the non-trivial computational cost of fine-tuning LLMs, all reported results in our core evaluations are averaged over 3 random seeds with different initializations—a standard protocol in parameter-efficient fine-tuning literature. To rigorously assess the statistical reliability of GO-LoRA's observed improvements, we report both p-values from paired t-tests and the mean $\pm$ standard deviation across these runs.

Table 14 details the p-values for GO-LoRA compared against the baselines on key benchmarks.

Table 14: Statistical significance testing (paired t-test, 3 seeds) for GO-LoRA improvements.

Benchmark	**GO-LoRA vs. LoRA**	**GO-LoRA vs. DoRA**	**Significant** ($p < 0.05$)
LLaMA2-7B Commonsense	$p = 0.015$	$p = 0.020$	✓
LLaMA3-8B Commonsense	$p = 0.013$	$p = 0.011$	✓
LLaMA2-7B Math & Code	$p = 0.012$	$p = 0.008$	✓
Image Cls (ViT-B/16 Avg.)	$p = 0.016$	$p = 0.028$	✓
VL-BART (Avg.)	$p = 0.025$	$p = 0.044$	✓

We additionally detail the exact numerical bounds (Mean $\pm$ Std) for representative benchmarks:

- **LLaMA2-7B Commonsense**: GO-LoRA 82.67 ± 0.24, DoRA 81.00 ± 0.31, LoRA 80.76 ± 0.38.
- **LLaMA3-8B Commonsense**: GO-LoRA 87.05 ± 0.18, DoRA 85.16 ± 0.27, LoRA 85.27 ± 0.35.
- **LLaMA2-7B Math & Code**: GO-LoRA 39.83 ± 0.21, DoRA 36.61 ± 0.28, LoRA 37.77 ± 0.34.
- **Image Cls (ViT-B/16 Avg.)**: GO-LoRA 90.28 ± 0.30, DoRA 88.33 ± 0.36, LoRA 87.39 ± 0.46.
- **VL-BART Avg.**: GO-LoRA 78.01 ± 0.31, DoRA 77.38 ± 0.42, LoRA 76.51 ± 0.50.

The narrow standard deviations coupled with $p < 0.05$ on the representative benchmarks in Table 14 indicate that GO-LoRA's performance gains are robust to random initializations and statistically significant on these evaluations.

To further assess the reliability of improvements over the strongest competing methods, we conduct paired significance tests against StelLA and HiRA on LLaMA2-7B and LLaMA3-8B commonsense reasoning. Using three matched random seeds, we compute the eight-task average for each run and perform two-sided paired t-tests on the resulting aggregate scores. Table 15 reports the paired improvements, confidence intervals, and corresponding p-values.

Table 15: Paired significance testing against strong baselines on commonsense reasoning (3 matched seeds).

Backbone	Baseline	Δ	95% CI	p-value
LLaMA2-7B	StelLA	+0.32	[+0.07, +0.57]	0.031
LLaMA2-7B	HiRA	+1.24	[+0.44, +2.04]	0.022
LLaMA3-8B	StelLA	+0.32	[+0.11, +0.53]	0.023
LLaMA3-8B	HiRA	+0.33	[+0.06, +0.60]	0.035

GO-LoRA achieves positive paired improvements over both StelLA and HiRA across the two backbones, with all reported confidence intervals excluding zero. In particular, the $+0.32$-point improvements over StelLA are supported by paired p-values of 0.031 and 0.023 on LLaMA2-7B and LLaMA3-8B, respectively. These results provide additional statistical evidence for the consistent aggregate performance advantage of GO-LoRA over strong geometry-aware PEFT baselines.

J Future Work

A natural avenue for future work lies in extending GO-LoRA's manifold-aware parameterization to broader geometric structures beyond the hypersphere. Key directions include: (1) generalizing the geodesic construction to Stiefel and Grassmann manifolds, which may better capture the joint column-space structure of multi-head attention weights; (2) integrating GO-LoRA with quantization-aware fine-tuning methods such as QLoRA [24] and QA-LoRA to investigate whether manifold-constrained updates remain beneficial under extreme precision reduction; (3) exploring adaptive allocation strategies [78, 67, 12] within the tangent-space framework, where orthogonal projection may help inform dynamic rank pruning decisions based on effective rank utilization; and (4) applying GO-LoRA to emerging architectures such as Mixture-of-Experts (MoE) models [79] and state-space models, where the interaction between manifold geometry and sparse activation patterns remains largely unexplored.

K Limitations

While GO-LoRA demonstrates consistent improvements across diverse modalities, several directions remain worthy of further investigation. The current implementation adopts a fixed adaptation rank within each experimental setting to ensure controlled and comparable evaluations; however, adaptive rank allocation across layers or modules may further improve parameter efficiency under heterogeneous task requirements. In addition, our theoretical analysis primarily focuses on the local geometry induced by normalized directional updates and tangent-space reparameterization, while a more comprehensive characterization of global training dynamics under large update regimes, strong optimization noise, or low-precision training remains an interesting direction. Finally, although our experiments cover language understanding, mathematical reasoning, code generation, visual recognition, and multimodal reasoning, extending GO-LoRA to additional architectures and generative paradigms, such as pure encoder models, diffusion-based generative models [38], and larger-scale foundation models, may provide further insights into the generality of geometry-aware low-rank adaptation.

L Broader Impacts

GO-LoRA advances parameter-efficient fine-tuning by providing a geometry-aware adaptation framework, potentially enabling more effective model customization under limited parameter budgets. This may broaden access to large-model adaptation for researchers and practitioners with constrained hardware resources, and may reduce the compute required to reach a given level of adaptation quality.

However, techniques that improve the adaptability of large foundation models also carry inherent risks. More efficient fine-tuning could lower barriers for misuse in sensitive areas such as generating misleading content or developing biased decision systems. Importantly, GO-LoRA does not inherently mitigate biases that may be present in the original pre-trained models or the fine-tuning data. By concentrating the limited parameter budget more effectively on directional shifts, such biases could potentially be transferred or amplified more efficiently if not proactively identified and addressed. We therefore encourage practitioners to combine GO-LoRA with established bias auditing and fairness evaluation protocols when deploying adapted models in high-stakes applications.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the motivation, method, and main contributions of the paper, and these claims are supported by the theoretical analysis in Section 3 and the experiments in Section 4.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: A dedicated Limitations section discusses the main limitations of the current study, including fixed rank choices, module coverage, theoretical scope, and architecture coverage.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The paper states the assumptions for Proposition 1 and Theorems 2–3 and provides complete proofs for them in the appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper and appendix provide the backbone models, datasets, hyperparameters, optimizer settings, and task-specific experimental configurations, and an anonymized code repository is provided for reproducing the reported results.

Guidelines:

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

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: An anonymized code repository is provided for this submission, and all datasets used in the paper are publicly available and cited in the manuscript.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper and appendix provide the backbone models, target modules, optimizer settings, learning rates, schedules, batch sizes, and task-specific hyperparameters needed to understand the experiments.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The paper reports results averaged over three random seeds and provides paired t-tests together with mean $\pm$ standard deviation in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We summarize our computational resources in Appendix G and Appendix H.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research described in this paper complies with the NeurIPS Code of Ethics, and does not involve human-subject experimentation.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

Justification: A dedicated Broader Impacts section discusses both the potential positive impact of more efficient model adaptation and the possible negative impacts related to misuse, bias transfer, and deployment in high-stakes settings.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: This work does not release a high-risk pre-trained model, image generator, or scraped dataset, so this question is not directly applicable to the submission.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We have used open-source models and datasets, and have appropriately cited them.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

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

13. **New assets**

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

    Answer: [N/A]

    Justification: We use and cite existing datasets in this work

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing, user studies, or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

Justification: This work does not involve human-subject research, so IRB approval or an equivalent review was not required.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: [N/A]

Guidelines:

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