OREA: ORTHOGONAL RE-FACTORIZATION FOR LOW-RANK ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-rank adaptation is a prominent paradigm for the parameter-efficient fine-tuning of large-scale pre-trained models. Conventional LoRA updates weight as $\Delta W = BA$. Based on SVD, recent works parameterize the weight update in a three-factor form: $\Delta W = USV^\top$. Existing methods differ in how orthogonality is incorporated, ranging from initialization and regularization to manifold-based optimization. In this paper, we propose OReA, which restores orthogonality through a product-preserving QR–SVD re-factorization at designated training steps. Moreover, we propose a nonlinear variant algorithm, OReA+, to augment the representation with a nonlinear complementary branch. We establish the shared rank-r approximation limit of the linear forms with a common frozen reference and bounds for the structured nonlinear extension. Extensive experiments across diverse downstream tasks validate that our method consistently outperforms state-of-the-art methods. Code is available at `https://anonymous.4open.science/r/OReA-2BE9`.

1 INTRODUCTION

The advent of large-scale pre-trained models has fundamentally transformed the landscape of artificial intelligence in natural language processing, computer vision, and multimodal tasks (Awais et al., 2025; Zhou et al., 2025; He et al., 2022). We usually need to fine-tune the pre-trained models for downstream tasks. However, full Fine-Tuning is computationally prohibitive and storage-intensive for downstream applications (Raffel et al., 2020). Consequently, Parameter-Efficient Fine-Tuning (PEFT) has emerged as a critical research direction, aiming to adapt foundation models to specific tasks by updating only a minimal subset of parameters (Han et al., 2024; Hu et al., 2022).

Among PEFT paradigms, Low-Rank Adaptation (LoRA) (Hu et al., 2022) is widely used due to its simplicity and efficiency. LoRA hypothesizes that the weight updates during adaptation could be represented as $\Delta W = BA$, where B and A are low-rank matrices. However, the standard parameterization, initialized with Gaussian noise and zeros, often suffers from slow convergence and suboptimal local minima (Hayou et al., 2024). Furthermore, the coupling of magnitude and direction in the matrices B and A complicates the optimization landscape, potentially limiting the model's ability to learn diverse feature subspaces(represented by the columns of B and A) effectively (Li et al., 2025a; Zhang et al., 2023).

These limitations naturally promote us to analyze the fundamental principles of matrix approximation. SVD-inspired three-factor adaptation has been explored in prior work, including TriLoRA for image personalization (Feng et al., 2024), PoLAR (Lion et al., 2025), and StelLA (Li et al., 2025a). Recent geometric optimizers also study low-rank adaptation using fixed-rank and quotient-manifold formulations (Bogachev et al., 2026; ZHENG & Wu, 2026). These approaches motivate a precise question: **Can structured re-factorization restore orthogonal directions and a diagonal core without changing the intermediate linear update matrix?**

We address this question with Orthogonal Re-factorization for Adaptation (OReA), which restores an orthogonal three-factor representation through a product-preserving QR–SVD procedure. After an unconstrained update of U, S, V, a coupled QR–SVD procedure restores column orthogonality and diagonalizes the core while preserving the intermediate product. This separates factor optimization from structural restoration. OReA remains a rank-r linear representation. To extend the representation,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

OReA+ adds a complementary branch whose bases are derived nonlinearly from U and V. Our contributions can be summarized as follows:

- **Product-preserving structured re-factorization.** We develop a coupled QR–SVD update that restores column-orthogonal factors and a diagonal core while preserving the linear weight product, augmented by a two-sided orthogonal-complement nonlinear branch with only r^2 additional trainable parameters and a practical two-phase training schedule.
- **Theoretical Guarantees.** We establish a theoretical framework providing the lower bound on the approximation error. We establish a shared rank-r limit under a common frozen reference and two-sided bounds for the nonlinear family, with additional conditions required to attain its lower bound.
- **Comparison of Empirical Experiments.** Extensive experiments across various downstream tasks validate the effectiveness of our approach. OReA and OReA+ outperform state-of-the-art PEFT methods. Additional analyses characterize the dependence of performance on the task, learning rate, and rank–scaling configuration.

2 Related Work

Parameter-Efficient Fine-Tuning (PEFT). PEFT methods adapt pre-trained models with a small number of trainable parameters. Adapter modules (Hu et al., 2023; Han et al., 2024; Houlsby et al., 2019) and prompt tuning (Lester et al., 2021; Li & Liang, 2021; Potlapalli et al., 2023; Li et al., 2025b) insert task-specific components, while LoRA (Hu et al., 2022) injects low-rank update matrices $\Delta W = BA$ and has become a dominant baseline; recent work refines its training stability and expressiveness.

SVD-aware and Orthogonal Variants. PiSSA (Meng et al., 2024) and EDoRA (Nasiri & Garraghan, 2025) use SVD or QR of the pre-trained weight for initialization, but the spectral structure is not preserved during training; LoRAPrune (Zhang et al., 2024) and SVFT (Lingam et al., 2024) fine-tune within selected spectral subspaces. AdaLoRA (Zhang et al., 2023) and CLoRA (Lu et al., 2025) enforce approximate orthogonality through regularization, which does not yield strict orthogonality. Manifold-based methods (Liu et al., 2024b; Ding et al., 2023; Li et al., 2025a; Chen et al., 2026; Qiu et al., 2023; Feng et al., 2025; Bian et al., 2025) optimize on the Stiefel manifold; among geometric approaches, StelLA (Li et al., 2025a) adopts a three-factor $USV^\top$ form and maintains orthogonal factors using Riemannian gradient transformations and retraction.

Three-Factor LoRA Architectures. Decomposing the weight update into three matrices explores the trade-off between expressiveness and efficiency. LoRA-XS (Bałazy et al., 2024) utilizes a three-factor form ($R \times r \times R$) but keeps the projection matrices frozen to maximize parameter efficiency at the cost of adaptability. Other variants, such as ResLoRA (Shi et al., 2024), introduce residual paths within the low-rank structure. StelLA (Li et al., 2025a) proposes a Riemannian manifold-based method to optimize the three-factor form ($\Delta W = USV^\top$). OFT (Qiu et al., 2023) uses constrained orthogonal fine-tuning that imposes an additional radius constraint to the hypersphere. OMoE trains experts in an orthogonal manner to promote diversity (Feng et al., 2025). PoLAR also separates two directional factors from a dense scale matrix (Lion et al., 2025). In contrast to the dense cores of PoLAR and StelLA, OReA restores a diagonal core through coupled QR–SVD re-factorization.

Nonlinear Low-Rank Extensions. LoRAN (Li et al., 2024), AuroRA (Dong et al., 2025), and PEANuT (Zhong et al., 2025) introduce nonlinearity through activations, dynamic ranks, or auxiliary networks. DyLoRA (Valipour et al., 2023) instead studies training across multiple adapter ranks. Unlike AuroRA, the nonlinear branch in OReA+ is mergeable into W_0 at inference since no activation is applied to the input. More information can be seen in Appendix C.

Positioning of Our Work. OReA restores column-orthogonal U, V and a diagonal S through a coupled QR–SVD re-factorization that preserves the intermediate linear matrix product. OReA+ further augments this with an orthogonal-complement nonlinear branch that is mergeable into W_0 at inference, keeping the parameter budget close to standard LoRA.

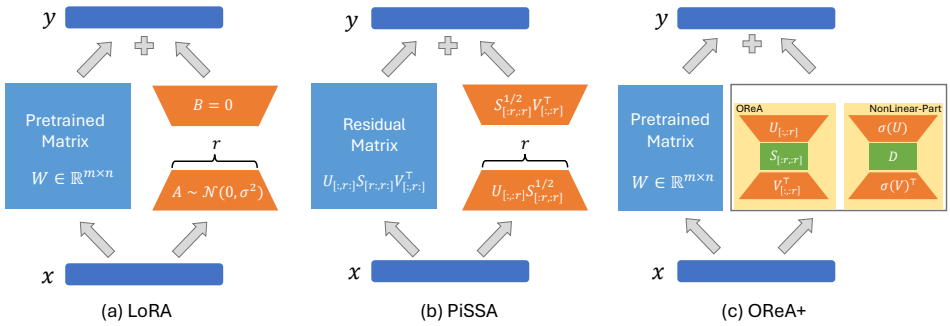

Figure 1: **Framework of our methods.** OReA restores column orthogonality and core diagonality via QR–SVD re-factorization. OReA+ adds a derived nonlinear branch in complementary subspaces.

3 Methodology

As depicted in Figure 1, we propose OReA and OReA+, a novel extension of the LoRA framework.

3.1 Preliminary on LoRA

Let $W \in \mathbb{R}^{d\times k}$ denote a weight matrix within a pre-trained neural network. LoRA parameterizes the weight updates during fine-tuning as $W = W_0 + \Delta W$, where W_0 represents the frozen pre-trained weights and $\Delta W \in \mathbb{R}^{d\times k}$ represents the learnable accumulation. To optimize parameter efficiency, LoRA approximates the update as $\Delta W = B \cdot A$, where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ are low-rank matrices with rank $r \ll r_{\max} = \min(d, k)$. Consequently, the forward pass for an input vector $\mathbf{x} \in \mathbb{R}^k$ is formulated as:

$$\mathbf{y} = W_0 \cdot \mathbf{x} + \Delta W \cdot \mathbf{x} = W_0 \cdot \mathbf{x} + B \cdot A \cdot \mathbf{x} \tag{1}$$

Challenges: LoRA exhibits two primary limitations: (a) **Insufficient Expressiveness:** Constrained by its linear low-rank structure, LoRA often necessitates a higher rank r to capture complex weight updates, thereby compromising parameter efficiency. (b) **Limited Optimization Flexibility:** There exist strong interdependencies and insufficient exploitation of the geometric structure underlying low-rank representation for B and A (Zhang et al., 2023). OReA addresses the update procedure, whereas OReA+ augments the representation with an additional complementary branch.

Though variants such as LoRA+ and PiSSA attempt to mitigate this by adjusting initialization schemes or learning rates, these underlying optimization challenges persist.

3.2 OReA: Product-Preserving Orthogonal Re-Factorization

Following SVD-inspired three-factor adaptation, we represent the linear weight update as

$$\Delta W = USV^{\top}, \qquad \mathbf{y} = W_0 \cdot \mathbf{x} + USV^{\top}\mathbf{x} \tag{2}$$

where $U \in \mathbb{R}^{d\times r}$ and $V \in \mathbb{R}^{k\times r}$ are column-orthogonal matrices, i.e., $U^{\top}U = I$ and $V^{\top}V = I$, and $S \in \mathbb{R}^{r\times r}$ is a diagonal matrix.

Directly applying SGD to update U, S, and V poses a challenge, as unconstrained gradient steps may violate the orthogonality constraints on U and V. Consequently, the implementation of OReA requires addressing two key issues: (1) appropriate initialization of the three factors, and (2) restoring orthogonality at the designated re-factorization steps. Inspired by SVD, we initialize U and V using the SVD of the pre-trained weights W_0, while initializing S to zero. We refer readers to PiSSA (Meng et al., 2024) for further details on SVD-based initialization. To maintain the structural constraints during training, we propose a two-step update mechanism. We first perform a standard unconstrained SGD update and then retract the resulting factorization back to the orthogonality-constrained form through a QR-SVD re-orthogonalization procedure.

For clarity, we express the unconstrained factor update using SGD:

$$U_{t+\frac{1}{2}} = U_t - \eta \nabla_{U_t} \mathcal{L}, \qquad V_{t+\frac{1}{2}} = V_t - \eta \nabla_{V_t} \mathcal{L}, \qquad S_{t+\frac{1}{2}} = S_t - \eta \nabla_{S_t} \mathcal{L}, \tag{3}$$

where η denotes the learning rate and $\mathcal{L}$ is the training loss. Here, $U_{t+\frac{1}{2}}$, $V_{t+\frac{1}{2}}$, and $S_{t+\frac{1}{2}}$ denote the intermediate factors obtained after the unconstrained SGD step. For the diagonal factor S, the update is applied only to its diagonal entries. Since $U_{t+\frac{1}{2}}$ and $V_{t+\frac{1}{2}}$ may no longer satisfy the column-orthogonality constraints, we subsequently restore their structure through a QR-SVD re-orthogonalization procedure.

We first compute QR factorizations of the intermediate factors:

$$U_{t+\frac{1}{2}} = Q_{U_{t+\frac{1}{2}}} \cdot R_{U_{t+\frac{1}{2}}}, \qquad V_{t+\frac{1}{2}} = Q_{V_{t+\frac{1}{2}}} \cdot R_{V_{t+\frac{1}{2}}}, \tag{4}$$

Next, we construct an intermediate matrix $P_{t+\frac{1}{2}} = R_{U_{t+\frac{1}{2}}} S_{t+\frac{1}{2}} R_{V_{t+\frac{1}{2}}}^\top$ and perform SVD on $P_{t+\frac{1}{2}}$ to re-diagonalize the core factor while preserving the overall update matrix:

$$P_{t+\frac{1}{2}} = R_{U_{t+\frac{1}{2}}} \cdot S_{t+\frac{1}{2}} \cdot R_{V_{t+\frac{1}{2}}}^\top = P_U \cdot P_S \cdot P_V^\top \tag{5}$$

where P_U and P_V are orthogonal matrices, and P_S is a diagonal matrix.

Whenever the QR–SVD re-factorization is applied, strict orthogonality of U and V, as well as the diagonal structure of S, are guaranteed by the following update rule:

$$U_{t+1} = Q_{U_{t+\frac{1}{2}}} \cdot P_U, \qquad V_{t+1} = Q_{V_{t+\frac{1}{2}}} \cdot P_V, \qquad S_{t+1} = P_S, \tag{6}$$

Lemma 1 (Product-preserving re-factorization). In exact arithmetic, Eqs. 4–6 produce column-orthogonal U_{t+1}, V_{t+1} and diagonal S_{t+1}, while satisfying $U_{t+1} S_{t+1} V_{t+1}^\top = U_{t+\frac{1}{2}} S_{t+\frac{1}{2}} V_{t+\frac{1}{2}}^\top$. The structural guarantees hold after each re-factorization; floating-point implementations satisfy them up to numerical precision. (We provide the proof in Appendix D.1.)

Remark 1. The QR–SVD step re-expresses the intermediate linear update matrix rather than projecting it to a different matrix. Its product-preservation property does not imply equivalence to gradient descent on the full weight matrix, nor does it imply preservation of future optimizer trajectories or adaptive optimizer states. Manifold-based methods such as StelLA (Li et al., 2025a) use a different geometric update rule.

3.3 OREA+: A NONLINEAR VARIANT OF OREA

The formalism of OReA mirrors a truncated SVD of ΔW. Consequently, its expressive capacity is still limited by a linear rank-r subspace approximation. To alleviate this limitation, we propose a nonlinear variant, denoted as OReA+. Here, "nonlinear" refers to the mapping applied to the factor matrices U and V, rather than to the input activations. We introduce a matrix-wise nonlinear mapping $\sigma(\cdot) : \mathbb{R}^{m \times r} \to \mathbb{R}^{m \times r}$ that preserves the input shape:

$$\Delta W = U \cdot S \cdot V^\top + \sigma(U) \cdot D \cdot \sigma(V)^\top \tag{7}$$

where $D \in \mathbb{R}^{r \times r}$ is a learnable parameter matrix. While various nonlinear functions can be employed for the intermediate mapping $f(\cdot)$, such as the Hadamard product $f(\mathbf{X}) = \mathbf{X} \odot \mathbf{X}$, it is crucial to reduce feature redundancy between the linear and nonlinear branches. Specifically, to reduce dependencies between the linear basis U and its nonlinear counterpart $\sigma(U)$, we construct $\sigma(U)$ by projecting a nonlinear transformation onto the orthogonal complement of U. Similarly, $\sigma(V)$ is constructed with respect to V:

$$\begin{aligned} \sigma(U) &= \mathrm{Norm}(f(U) - U \cdot U^\top \cdot f(U)) \\ \sigma(V) &= \mathrm{Norm}(f(V) - V \cdot V^\top \cdot f(V)) \end{aligned} \tag{8}$$

where $f(\cdot)$ is an arbitrary nonlinear function, e.g., $f(U) = U \odot U$ in our experiments, and $\mathrm{Norm}(\cdot)$ denotes column-wise L_2-normalization. This construction makes the nonlinear bases orthogonal

to the original linear bases. However, the nonlinear bases themselves are not necessarily mutually column-orthogonal. For the theoretical analysis, a zero residual column is defined to remain zero under normalization. The nonlinear features need not have full column rank.

Lemma 2: If U and V are column-orthogonal matrices, then $\sigma(U)$ and $\sigma(V)$ satisfy the orthogonality conditions $\sigma(U)^\top \cdot U = 0$ and $\sigma(V)^\top \cdot V = 0$. (We provide the proof in Appendix D.2.)

To optimize the parameters of OReA+, we consider the general joint-training case where U, S, V, and D are updated together. We employ standard SGD, followed by the orthogonality steps described in Eqs. 4–6 to maintain the column-orthogonal property of the factors U and V. However, applying these re-orthogonalization operations changes U and V, which in turn changes the nonlinear bases $\sigma(U)$ and $\sigma(V)$ in the second term of Eq. 7. To reduce the resulting discrepancy, we introduce a least-squares correction procedure for updating D.

After updating U, S, V, and D via SGD, we denote the intermediate nonlinear bases as $\sigma(U_{t+\frac{1}{2}})$ and $\sigma(V_{t+\frac{1}{2}})$. Let $\sigma(U_{t+1})^+$ and $\sigma(V_{t+1})^+$ denote the Moore-Penrose pseudo-inverses of $\sigma(U_{t+1})$ and $\sigma(V_{t+1})$, respectively. We adjust D as:

$$D_{t+1} = \big(\sigma(U_{t+1})^+ \sigma(U_{t+\frac{1}{2}})\big) D_{t+\frac{1}{2}} \cdot \big(\sigma(V_{t+1})^+ \sigma(V_{t+\frac{1}{2}})\big)^\top \tag{9}$$

where $D_{t+\frac{1}{2}} = D_t - \eta \nabla_{D_t} \mathcal{L}$, η is the learning rate, and $\mathcal{L}$ is the training loss.

Lemma 3: The correction in Eq. 9 provides a least-squares adjustment of D that minimizes the discrepancy in the nonlinear contribution before and after the re-orthogonalization step. This correction is exact only if the pre-re-factorization nonlinear matrix can be represented in the new pair of nonlinear subspaces. Otherwise, it gives the closest matrix in those subspaces under the Frobenius norm; it does not preserve the nonlinear contribution exactly. (We provide the proof in Appendix D.3.)

3.4 The Proposed Algorithm

The detailed training procedure of OReA+ is summarized in Algorithm 1 in Appendix A. We initialize U and V as column-orthogonal factors from the SVD of the pre-trained weight matrix W_0, and initialize both S and D as zero. In the general joint-training formulation, each iteration consists of four steps: (i) computing the training loss based on $W_0 + U_t S_t V_t^\top + \sigma(U_t) D_t \sigma(V_t)^\top$; (ii) applying an unconstrained SGD update to obtain the intermediate factors; (iii) restoring the column-orthogonal structure of U and V and the diagonal structure of S through the QR-SVD re-orthogonalization procedure in Eqs. 4–6; and (iv) adjusting D via the least-squares correction in Eq. 9.

The correction in Eq. 9 is mainly required when U and V are updated together with D, since the nonlinear bases $\sigma(U)$ and $\sigma(V)$ change after the re-orthogonalization step. In practice, we use a two-phase schedule: during the first T_1 epochs, we train only the linear OReA factors U, S, V with $D = 0$; afterwards, we freeze U, S, V and train only D for the nonlinear branch. Under this two-phase schedule, $\sigma(U)$ and $\sigma(V)$ remain fixed in the second phase, so the least-squares correction of D is not required at every step, saving much computational time. Since the second phase freezes U, S, V and optimizes only D, any improvement over the phase-one checkpoint reflects the incremental benefit of training the nonlinear complementary branch. Assessing this incremental benefit does not require the linear phase to have converged.

Remarks. (i) OReA can be recovered by removing the nonlinear branch, i.e., setting $D = 0$. (ii) The QR-SVD re-orthogonalization can be performed periodically to reduce overhead. (iii) Since the nonlinearity acts on the factors rather than the input activations, the update $USV^\top + \sigma(U) D \sigma(V)^\top$ can be merged into W_0 for inference.

Re-factorization frequency. Let q denote the number of optimizer steps between QR–SVD re-factorizations. The structural result in Lemma 1 applies at each re-factorization. When $q > 1$, it does not assert exact orthogonality at intermediate steps. The reported commonsense configuration uses $q = 100$, as specified in Appendix F. The orthogonal-complement interpretation of the nonlinear branch requires orthonormal factors when its bases are constructed.

3.5 THEORETICAL ANALYSIS

We analyze the lower bound of the approximation error of OReA and OReA+. Let $\Delta W^* \in \mathbb{R}^{d\times k}$ be the optimal update with SVD $\Delta W^* = U^* S^* {V^*}^\top$, $S^* = \mathrm{diag}(s_1, \ldots, s_{r_{\max}})$ with $|s_1| \geq \cdots \geq |s_{r_{\max}}|$ and $r_{\max} = \min(d, k)$. Proofs are in Appendix D.

Lemma 4 (OReA): For $U^\top U = V^\top V = I$ and diagonal S,

$$\min_{U,S,V} \|\Delta W^* - USV^\top\|_F^2 = \sum_{i=r+1}^{r_{\max}} s_i^2. \tag{10}$$

This is a best-representation result, not a convergence guarantee for a particular optimizer. (We provide the proof in Appendix D.4.)

The same representation bound holds for standard LoRA BA and for an unconstrained-core three-factor form when they represent all matrices of rank at most r relative to the same frozen weight. This does not imply identical training dynamics. PiSSA (Meng et al., 2024) instead freezes a residual weight after extracting the pretrained principal component; its rank-r trainable component is therefore defined relative to a different frozen reference.

Lemma 5 (OReA+): Let $m = \min(2r, r_{\max})$ and define $E_+^* = \inf_{U,S,V,D} \|\Delta W^* - USV^\top - \sigma(U)D\sigma(V)^\top\|_F^2$, where U, V are column-orthogonal and S is diagonal. Then $\sum_{i=m+1}^{r_{\max}} s_i^2 \leq E_+^* \leq \sum_{i=r+1}^{r_{\max}} s_i^2$. An update attains the lower bound only if it represents a best rank-m approximation of ΔW^*. Alignment of the combined left and right spans alone does not ensure this representability. (We provide the proof in Appendix D.5.)

Combining Lemmas 4 and 5, the nonlinear family contains the linear rank-r family and is contained in the rank-at-most-$2r$ family. Whether it strictly improves the approximation depends on the target and the nonlinear features. OReA differs from the compared linear methods in its factor-update and re-factorization procedure rather than in the global rank-r approximation limit; see Appendix C.

4 EXPERIMENTS

We evaluate OReA and OReA+ across three settings: synthetic study, commonsense reasoning, and natural language understanding. We further analyze hyperparameter sensitivity, ablations and computational overhead to assess the robustness and efficiency of our methods.

4.1 EXPERIMENTAL SETUP

Datasets and Models. For the synthetic study, we generate target matrices $\Delta W^* \in \mathbb{R}^{128\times 512}$ with entries drawn from $\mathcal{U}[-10^{-4}, 10^{-4}]$, in either a full-rank or rank-4 ($\Delta W^* = PQ$, $P \in \mathbb{R}^{128\times 4}$, $Q \in \mathbb{R}^{4\times 512}$) configuration. For real-world evaluation, we use GLUE for natural language understanding and six commonsense reasoning benchmarks, statistics are in Appendix E. We additionally evaluate LLaMA3.1-8B on HellaSwag (Zellers et al., 2019) and WinoGrande (Sakaguchi et al., 2021). The additional two-benchmark results are reported separately from the original six-task evaluation. We evaluate our methods using a selection of representative pre-trained models, including RoBERTa-Base, LLaMA3.1-8B, and Qwen2.5-7B.

Baselines. We compare our methods against a range of fine-tuning methods, categorized into three groups: (1) full Fine-Tuning, (2) Standard LoRA, and (3) LoRA variants. The variants include LoRA+ (Hayou et al., 2024), DoRA (Liu et al., 2024a), AdaLoRA (Zhang et al., 2023), PiSSA (Meng et al., 2024), LoRAN (Li et al., 2024), StelLA (Li et al., 2025a), and CLoRA (Lu et al., 2025).

Implementation Details. To ensure reproducibility, we outline the general experimental setup here, with comprehensive hyperparameter configurations provided in Appendix F. We use standard LoRA fine-tuning protocols for the GLUE benchmarks. To ensure statistical reliability, we report the average results of five independent runs using distinct random seeds. A detailed analysis of training computational overhead is presented in Appendix G.7. We follow each baseline's recommended target-module configuration. OReA and OReA+ use the same target modules as StelLA. Trainable parameter counts are reported for each method in Table 1.

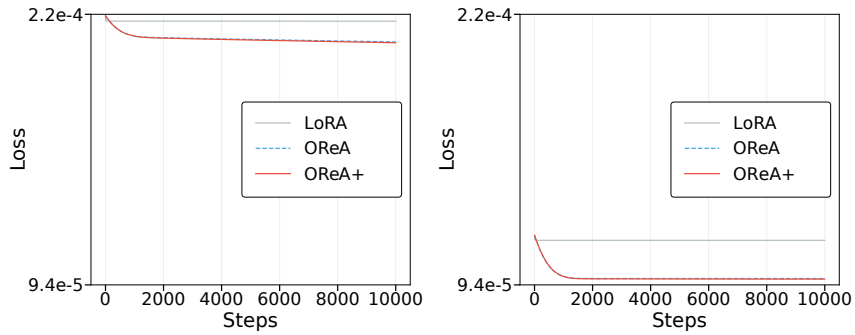

Figure 2: Training dynamics of different adaptation methods on the synthetic recovery task. We optimize the squared Frobenius recovery loss $\mathcal{L}_{\text{syn}} = \|\Delta W^* - \widehat{\Delta W}\|_F^2$, where $\widehat{\Delta W}$ denotes the update represented by each method. All methods use nominal adapter rank $r = 16$. A smaller loss is better. (a) ΔW^* is full rank. (b) ΔW^* has rank 4.

Table 1: Commonsense reasoning evaluation results for LLaMA3.1-8B and Qwen2.5-7B on six tasks. The best results are highlighted in bold. For all datasets, a higher value is better. Avg. (6) is the unweighted arithmetic mean of the six listed task scores.

Backbone	Method	# Trainable Params	ARC-C	ARC-E	BoolQ	OBQA	PIQA	SIQA	Avg. (6)
LLaMA 3.1-8B	LoRA	20.97M	52.90	72.64	67.12	65.60	65.83	67.20	65.22
	AdaLoRA	28.31M	59.56	76.14	62.29	65.00	83.95	72.67	69.94
	DoRA	22.35M	52.56	71.84	69.39	65.20	63.55	68.17	65.12
	LoRA+	20.97M	53.84	72.85	70.21	70.80	70.62	74.21	68.75
	PiSSA	20.97M	57.42	74.16	68.41	72.00	71.98	76.00	70.00
	LoRAN	20.97M	58.27	69.15	67.28	66.52	66.18	68.16	65.93
	CLoRA	20.97M	42.75	56.94	51.28	53.80	57.29	61.51	53.93
	StelLA	14.17M	70.65	90.24	71.28	81.60	86.18	**78.97**	79.82
	Ours(OReA)	14.17M	74.57	**91.25**	72.60	**82.20**	86.34	78.92	80.98
	Ours(OReA+)	14.17M	**77.30**	90.03	**72.63**	**82.20**	**87.81**	78.15	**81.35**
Qwen2.5-7B	LoRA	20.18M	79.61	88.00	64.74	82.60	84.60	76.97	79.42
	AdaLoRA	27.07M	85.84	93.35	60.31	83.40	86.89	76.82	81.10
	DoRA	21.58M	79.61	91.37	66.91	84.80	86.07	79.22	81.33
	LoRA+	20.18M	82.00	90.36	69.66	86.40	86.51	80.76	82.61
	PiSSA	20.18M	85.49	93.94	70.34	87.80	88.03	**80.96**	84.43
	LoRAN	20.18M	79.74	89.27	64.88	84.16	87.22	79.21	80.75
	CLoRA	20.18M	78.33	85.52	59.48	80.60	83.62	77.79	77.56
	StelLA	13.54M	81.31	94.70	71.38	86.40	87.65	80.14	83.60
	Ours(OReA)	13.54M	85.92	94.49	**71.96**	88.80	88.57	79.53	84.88
	Ours(OReA+)	13.54M	**86.86**	**95.03**	71.44	**90.20**	**88.68**	80.14	**85.39**

4.2 Synthetic Recovery Study

We conduct a controlled empirical study on synthetic data. We formulate synthetic recovery as directly approximating a target update matrix ΔW^*. For each method, let $\widehat{\Delta W}$ denote its represented update. We optimize the squared Frobenius recovery objective $\mathcal{L}_{\text{syn}} = \|\Delta W^* - \widehat{\Delta W}\|_F^2$. We consider two distinct regimes for ΔW^*: a full-rank setting and a low-rank setting where $rank(\Delta W^*) = 4$ (see dataset description part). All models–standard LoRA, OReA, and OReA+–are optimized via SGD under identical conditions. Figure 2 illustrates the training dynamics and final approximation errors. The key findings are summarized below.

Approximation quality. Both proposed variants consistently outperform standard LoRA in convergence speed and final loss value. In Fig. 2(a) (full-rank ΔW^*), with all methods using nominal adapter rank $r = 16$, when reaching a stable state, OReA achieves approximately a 6% lower loss than standard LoRA. OReA+ further improves upon OReA, attaining the lowest loss. A similar trend holds in the low-rank regime (Figure 2(b), rank$(\Delta W^*) = 4$), where OReA+ maintains its lead and standard LoRA exhibits the highest loss (approximately 11% higher than OReA+). Comparing Figure 2(a) and (b), the approximation loss for all methods decreases when the target ΔW^* is low-rank. The approximation error is $\leq 1 \times 10^{-4}$, meaning that standard LoRA is relatively sufficient for approximating simple (low-rank) updates. For the linear methods, LoRA and OReA share the

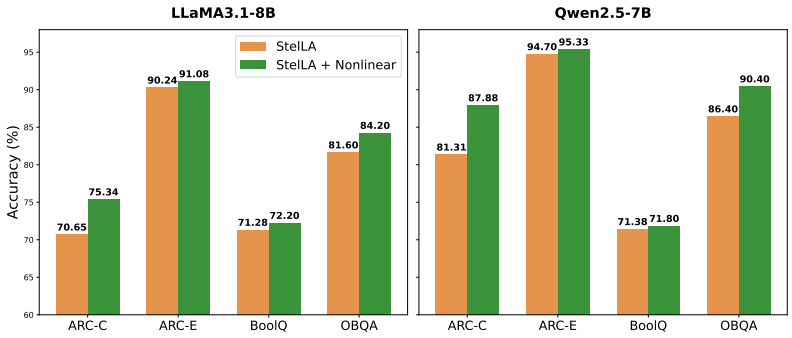

Figure 3: Performance gains when augmenting StelLA with our nonlinear branch (Eq.7). "StelLA" denotes the original method; "StelLA + Nonlinear" replaces the forward function with Eq.7.

same rank-r representation class, so their difference in achieved recovery loss reflects the tested optimization procedure rather than a larger linear hypothesis class. OReA+ additionally introduces a second branch and therefore has a larger structured representation family. Overall, the synthetic recovery results show that OReA and OReA+ provide more accurate approximations of target weight updates than LoRA, especially when the target update is high-rank.

4.3 Downstream NLP Evaluation

We design two tasks: Commonsense Reasoning and Natural Language Understanding. For the commonsense reasoning task, we select LLaMA3.1-8B and Qwen2.5-7B as the base models. The results are shown in Table 1. In the NLU task, we select RoBERTa-Base as the pretrained model and compare OReA and OReA+ with six recent widely-used fine-tuning methods across all four datasets of the GLUE benchmark.

Commonsense reasoning performance. The commonsense reasoning evaluation results are summarized in Table 1. OReA and OReA+ obtain competitive scores under the original configurations in Table 1. On LLaMA 3.1-8B, OReA+ achieves the highest scores on ARC-C (77.30%), BoolQ (72.63%), and PIQA (87.81%), while OReA attains the best performance on ARC-E (91.25%) and OBQA (82.20%). The advantages persist on Qwen2.5-7B, where OReA achieves the best results on BoolQ (71.96%), and OReA+ leads on ARC-C (86.86%), ARC-E (95.03%), OBQA (90.20%), and PIQA (88.68%). SIQA is the only task on which our methods do not attain the top score, but the gap is small and the best-performing baseline differs, suggesting that no single competing method dominates SIQA consistently. Overall, our methods achieve the best score on 10 of the 12 task–model combinations, demonstrating robust effectiveness across diverse model families. Figure 6 further examines learning-rate sensitivity. Across the tested learning-rate range, our methods attain the highest performance on both ARC-C and ARC-E.

Nonlinear extensions to LoRA variants. To evaluate the transferability of the proposed nonlinear complementary branch, we incorporate it into StelLA (Li et al., 2025a). Specifically, we augment StelLA's linear update with the complementary term in Eq. 7, while retaining its original optimization procedure for enforcing orthogonality. The branch introduces only r^2 additional trainable parameters per adapted weight matrix. As shown in Figure 3, the resulting StelLA + Nonlinear variant improves over the original StelLA on all task–model pairs shown. These results support the applicability of the proposed complementary branch beyond the OReA backbone.

OReA is competitive on NLU. In Table 7(in Appendix G.1), we analyze the performance of our methods on the GLUE benchmark using the RoBERTa-Base model. At $r = 8$ OReA (or OReA+) achieves the best PEFT result on three of four GLUE tasks—MRPC (88.97), STS-B (90.87), and CoLA (64.39, $+0.82$ over StelLA)—while trailing the top baseline by 0.39 on RTE. At $r = 2$, OReA+ leads on MRPC (89.21) but is behind StelLA by 0.20–1.10 on the other three tasks; the gap closes when r is increased to 8. Across all settings, every PEFT method lands within $\sim$1.5 points of full fine-tuning, so GLUE on RoBERTa-Base offers limited headroom to differentiate methods.

Table 2: Impact of hyperparameters (r and α) on ARC-C using method OReA+ with Qwen2.5-7B.

Rank r		**2**	**4**	**8**	**16**	**32**
$\alpha = 2r$	α	4	8	16	32	64
	Score	86.43	**87.80**	86.86	86.43	81.48
$\alpha \approx 5.6\sqrt{r}$	α	8	11	16	22	32
	Score	86.86	85.49	86.86	**87.03**	86.26
$\alpha \approx 2\sqrt{r}$	α	3	4	6	8	11
	Score	86.35	87.12	87.20	86.43	**88.48**

4.4 Sensitivity, Ablations, and Training Efficiency

We further analyze OReA and OReA+ in terms of learning-rate and rank–scaling sensitivity (Figures 6 and 5, and Table 2), effective rank of the learned updates (Appendix G.5), ablations of the re-factorization interval and nonlinear branch (Appendix G.6), and training efficiency (Appendix G.7).

Hyperparameter sensitivity. On ARC-E, Figure 5 shows that OReA leads at $r = 8, 16, 32$ under $\alpha \approx 2\sqrt{r}$, attaining its highest score at $r = 32$. On ARC-C, Table 2 shows that under the common heuristic $\alpha = 2r$, the highest accuracy (87.80%) is achieved at $r = 4$ ($\alpha = 8$), whereas increasing the rank to $r = 32$ ($\alpha = 64$) reduces the accuracy to 81.48%. We also explore alternative scaling strategies (second and third blocks). Competitive performance can be attained with different combinations. Taken together, these observations indicate that while a wide range of (r, α) configurations yields competitive performance, extreme settings can still cause notable degradation.

Effective rank of learned updates. To test whether the additional capacity of OReA+ is actually used by trained adapters, rather than only permitted by its rank-$2r$ bound, we examine the singular spectra of their learned updates. A final-checkpoint analysis on LLaMA3.1-8B ARC-E (Appendix G.5, Figure 7) shows that OReA+ has mean effective rank above r at every tested nominal rank, whereas LoRA remains below r.

Ablation analysis. We ablate two components of OReA+: the QR–SVD orthogonal re-factorization in its OReA backbone (with the nonlinear branch disabled) and the nonlinear branch activated after the OReA phase reaches a training-loss plateau. As shown in Appendix G.6 (Figure 8), ARC-E accuracy is highest at a re-factorization interval of $q = 10$ and declines modestly with sparser restoration, while enabling the nonlinear branch further lowers ARC-C training loss. We additionally evaluate $q = \infty$, corresponding to no QR–SVD re-factorization during training, which yields lower accuracy than finite re-factorization intervals and further supports the benefit of periodic structural restoration.

Training efficiency. We compare the training time between our method and the LoRA. We compare OReA with LoRA to carefully compare the difference of orthogonality constraints in Table 8 in the Appendix. Table 8 shows the additional cost is minimal, demonstrating the efficiency of our method.

5 Conclusion and Limitations

We propose OReA, which restores orthogonal factors and a diagonal core through QR–SVD re-factorization while preserving the intermediate linear update. OReA+ adds a mergeable nonlinear orthogonal-complement branch. Our analysis shows that OReA shares LoRA's rank-r approximation limit under a common frozen reference, whereas OReA+ expands the representation to a structured rank-at-most-$2r$ family, with additional conditions required to attain its lower approximation bound. Extensive experiments across diverse downstream tasks demonstrate that OReA and OReA+ consistently outperform state-of-the-art PEFT methods. Despite these results, OReA incurs an additional 0.06–$0.42\times$ training cost from QR–SVD re-factorization. Moreover, OReA+ uses a fixed nonlinear mapping, and our evaluation is limited to four GLUE tasks and 7–8B models on commonsense reasoning. Future work will explore cheaper optimization, alternative mappings, larger models, and generative tasks.

AI USE STATEMENT

In this work, we used generative AI tools for assist with translation. We have not used generative AI tools for help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, support qualitative and thematic data analysis, interpret results, and generate synthetic data sets, clean and reformat dataset are not applicable to this work. Additionally, we used generative AI tools for edit a research paper to improve readability. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundation models defining a new era in vision: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2245–2264, 2025. doi: 10.1109/TPAMI.2024.3506283.

Klaudia Bałazy, Mohammadreza Banaei, Karl Aberer, and Jacek Tabor. Lora-xs: Low-rank adaptation with extremely small number of parameters. *arXiv preprint arXiv:2405.17604*, 2024.

Fengmiao Bian, Jinyang ZHENG, Ziyun Liu, Jianzhou Luo, and Jian-Feng Cai. Finding low-rank matrix weights in DNNs via riemannian optimization: RAdagrad and RAdamw. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=tiGFiCrmKm.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*, 2019. URL https://api.semanticscholar.org/CorpusID:208290939.

Vladimir Bogachev, Vladimir Aletov, Alexander Molozhavenko, Denis Bobkov, Vera Soboleva, Aibek Alanov, and Maxim Rakhuba. LoRA meets riemannion: Muon optimizer for parametrization-independent low-rank adapters. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=WtbXgc9GVA.

Dexia Chen, Qianjie Zhu, Weibing Li, Yue Yu, Tong Zhang, and Ruixuan Wang. Preserve and sculpt: Manifold-aligned fine-tuning of vision-language models for few-shot learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZGJJF1e2u0.

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 Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *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)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

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, 2018. URL https://arxiv.org/abs/1803.05457.

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.

Haonan Dong, Wenhao Zhu, Guojie Song, and Liang Wang. AuroRA: Breaking low-rank bottleneck of loRA with nonlinear mapping. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2hgHyoyVWj.

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, 2024. URL https://arxiv.org/abs/2405.11236.

Jinyuan Feng, Zhiqiang Pu, Tianyi Hu, Dongmin Li, Xiaolin Ai, and Huimu Wang. Omoe: Diversifying mixture of low-rank adaptation by orthogonal finetuning, 2025. URL https://arxiv.org/abs/2501.10062.

G.H. Golub, Alan Hoffman, and G.W. Stewart. A generalization of the eckart-young-mirsky matrix approximation theorem. *Linear Algebra and its Applications*, 88-89:317–327, 1987. ISSN 0024-3795. doi: https://doi.org/10.1016/0024-3795(87)90114-5. URL https://www.sciencedirect.com/science/article/pii/0024379587901145.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=lIsCS8b6zj.

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*, ICML'24. JMLR.org, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15979–15988, 2022. doi: 10.1109/CVPR52688.2022.01553.

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

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5254–5276, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.319. URL https://aclanthology.org/2023.emnlp-main.319/.

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora, 2023. URL https://arxiv.org/abs/2312.03732.

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*, pp. 3045–3059, 2021.

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)*, pp. 4582–4597, 2021.

Yinqiao Li, Linqi Song, and Hanxu Hou. LoRAN: Improved low-rank adaptation by a non-linear transformation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3134–3143, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.177. URL https://aclanthology.org/2024.findings-emnlp.177/.

Zhizhong Li, Sina Sajadmanesh, Jingtao Li, and Lingjuan Lyu. StelLA: Subspace learning in low-rank adaptation using stiefel manifold. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=55Lv1unlUL.

Zongqian Li, Yixuan Su, and Nigel Collier. A survey on prompt tuning, 2025b. URL https://arxiv.org/abs/2507.06085.

Vijay Lingam, Atula Tejaswi, Aditya Vavre, Aneesh Shetty, Gautham Krishna Gudur, Joydeep Ghosh, Alex Dimakis, Eunsol Choi, Aleksandar Bojchevski, and Sujay Sanghavi. Svft: parameter-efficient fine-tuning with singular vectors. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Kai Lion, Liang Zhang, Bingcong Li, and Niao He. PoLAR: Polar-decomposed low-rank adapter representation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=jDxFD45kkc.

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 *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

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 *International Conference on Learning Representations*, volume 2024, pp. 38317–38350, 2024b.

Yuheng Lu, Bingshuo Qian, Caixia Yuan, Huixing Jiang, and Xiaojie Wang. Controlled low-rank adaptation with subspace regularization for continued training on large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19165–19181, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.940. URL https://aclanthology.org/2025.acl-long.940/.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: principal singular values and singular vectors adaptation of large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

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 Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Hamid Nasiri and Peter Garraghan. Edora: Efficient weight-decomposed low-rank adaptation via singular value decomposition. *arXiv preprint arXiv:2501.12067*, 2025.

Vaishnav Potlapalli, Syed Waqas Zamir, Salman Khan, and Fahad Shahbaz Khan. Promptir: prompting for all-in-one blind image restoration. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

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. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, August 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan

(eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454/.

Shuhua Shi, Shaohan Huang, Minghui Song, Zhoujun Li, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. ResLoRA: Identity residual mapping in low-rank adaption. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8870–8884, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.525. URL https://aclanthology.org/2024.findings-acl.525/.

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 Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3274–3287, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.239. URL https://aclanthology.org/2023.eacl-main.239/.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. LoRAPrune: Structured pruning meets low-rank parameter-efficient fine-tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3013–3026, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.178. URL https://aclanthology.org/2024.findings-acl.178/.

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. URL https://openreview.net/forum?id=lq62uWRJjiY.

Jinyang ZHENG and Tong Wu. LoRA-s: An efficient low rank adaptation scheme via sylvester equation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Guo2XGgxZA.

Yibo Zhong, Haoxiang Jiang, Lincan Li, Ryumei Nakada, Tianci Liu, Linjun Zhang, Huaxiu Yao, and Haoyu Wang. Peanut: Parameter-efficient adaptation with weight-aware neural tweakers, 2025. URL https://arxiv.org/abs/2410.01870.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *International Journal of Machine Learning and Cybernetics*, 16(12):9851–9915, 2025.

A Algorithm Details and Complexity

Complexity (per iteration). Beyond standard LoRA, the additional cost arises from: (i) the reconstruction of ΔW in Line 5, $O(drk + kr^2)$; (ii) QR and SVD decompositions in Line 8,

$O(dr^2 + kr^2)$ and $O(r^3)$, respectively; (iii) the nonlinear branch in Line 9, where computing $\sigma(U), \sigma(V)$ and the pseudo-inverse costs $O(dr^2 + kr^2)$, and solving Eq. 9 adds $O(dr^2 + kr^2 + r^3)$. Since $r \ll \min(d, k)$, the $O(r^3)$ term is negligible, giving a total additional per-iteration complexity of $O(drk + (d+k)r^2)$. The QR–SVD re-factorization itself costs $O((d+k)r^2 + r^3)$ per application. Explicit construction of the dense update incurs an additional $O(dkr)$ cost and should be distinguished from re-factorization. With interval q, the re-factorization cost is amortized over q optimizer steps.

Algorithm 1 General joint-training formulation of OReA+

1: **Input:** Pre-trained model, loss $\mathcal{L}$, rank r, iterations T.
2: **Output:** $\Delta W = USV^\top + \sigma(U)D\sigma(V)^\top$.
3: Initialize column-orthogonal U, V; set $S = 0$, $D = 0$.
4: **for** $t = 0 : (T-1)$ **do**
5: Compute $\mathcal{L}_t = \mathcal{L}(W_0 + U_t S_t V_t^\top + \sigma(U_t)D_t\sigma(V_t)^\top)$.
6: Compute gradients $\nabla_{U_t}, \nabla_{S_t}, \nabla_{V_t}, \nabla_{D_t}$.
7: SGD update to $U_{t+\frac{1}{2}}, S_{t+\frac{1}{2}}, V_{t+\frac{1}{2}}, D_{t+\frac{1}{2}}$ via Eq. 3.
8: Update $U_{t+1}, S_{t+1}, V_{t+1}$ via Eqs. 4–6.
9: Update D_{t+1} via Eq. 9.
10: **end for**
11: **return** $\Delta W = USV^\top + \sigma(U)D\sigma(V)^\top$.

The reported implementation instead uses a two-phase schedule. In the first phase, D is set to zero and only the linear factors U, S, V are trained. The AdamW optimizer state for U, S, V is retained across QR–SVD re-factorizations. In the second phase, U, S, V are frozen and only D is optimized.

B Comparison with Orthogonal/SVD-aware Variants

Our method distinguishes itself from prior column-orthogonal variants of LoRA in three respects.

(1) **CLoRA** (Lu et al., 2025) adds an orthogonal regularizer to the loss, which is a *soft* constraint and does not guarantee strict orthogonality.

(2) **StelLA** (Li et al., 2025a) optimizes on the Stiefel manifold, but discards the gradient component orthogonal to the tangent space.

(3) **PiSSA** (Meng et al., 2024) enforces orthogonality only at initialization; the spectral structure is not preserved during training. In contrast, OReA restores column orthogonality and a diagonal core at the designated QR–SVD re-factorization steps.

C Comparison of Representative PEFT Methods

Table 3 compares representative PEFT methods from four perspectives: the factorization form of the weight update ΔW, the mechanism used to encourage or enforce orthogonality, whether an additional nonlinear branch is introduced, and the number of trainable parameters for an adapted weight matrix of size $d \times k$ with adapter rank r.

The compared methods exhibit different design choices. Standard LoRA uses a two-factor low-rank decomposition and does not impose orthogonality constraints. Several subsequent methods improve LoRA from different angles. For example, PiSSA initializes the low-rank factors from the principal singular components of the pre-trained weight matrix, but the orthogonality of the initialized factors is not explicitly preserved during training. CLoRA uses a regularization-based constraint, whereas StelLA maintains orthogonal factors through manifold-based optimization, but their orthogonality control is either soft or not coupled with a strict SVD-like factorization throughout optimization.

In contrast, OReA explicitly parameterizes the update as a three-factor decomposition, $\Delta W = USV^\top$, where U and V are column-orthogonal and S is diagonal. The QR–SVD re-factorization restores this structure at the designated steps, thereby separating the directional factors from the scaling factor. This makes OReA structurally closer to an SVD-style adaptation than standard two-factor LoRA variants.

Table 3: Comparison of representative PEFT methods.

Method	Factors	Orthogonality	Nonlinear	# Trainable Params
LoRA (Hu et al., 2022)	2 (B, A)	None	No	$r(d+k)$
AdaLoRA (Zhang et al., 2023)	3 (P, Λ, Q)	Soft (regularization)	No	$\approx (r{+}1)(d{+}k)$
DoRA (Liu et al., 2024a)	2 (B, A) + magnitude	None	No	$r(d+k)+d$
LoRA+ (Hayou et al., 2024)	2 (B, A)	None	No	$r(d+k)$
PiSSA (Meng et al., 2024)	2 (B, A)	SVD init only	No	$r(d+k)$
LoRAN (Li et al., 2024)	2 (B, A)	None	Yes	$r(d+k)$
CLoRA (Lu et al., 2025)	2 (B, A)	Soft (regularization)	No	$r(d+k)$
StelLA (Li et al., 2025a)	3 (U, S, V)	Riemannian GD	No	$r(d{+}k)+r^2$
OReA (ours)	3 (U, S, V)	QR+SVD retraction	No	$r(d{+}k)+r$
OReA+ (ours)	3 + nonlinear (D)	QR+SVD retraction	Yes	$r(d{+}k)+r+r^2$

OReA+ further extends this formulation by adding a nonlinear branch constructed in the orthogonal complement of the linear bases. This branch increases the expressive capacity of the update while preserving the mergeability of the learned adaptation at inference time. Importantly, the additional trainable parameters remain modest: OReA uses $r(d+k)+r$ parameters, while OReA+ uses $r(d+k)+r+r^2$, which remains close to the budget of standard LoRA when r is small.

D Theoretical Proofs

D.1 Proof of Lemma 1

Proof. We begin with the intermediate effective weight update after the unconstrained SGD step in Eq. 3:

$$\Delta W_{t+1} = U_{t+\frac{1}{2}} S_{t+\frac{1}{2}} V_{t+\frac{1}{2}}^\top. \tag{11}$$

Substituting Eq. 4 and Eq. 5 into the Eq. 11, we have:

$$\begin{aligned}
\Delta W_{t+1} &= (Q_{U_{t+\frac{1}{2}}} R_{U_{t+\frac{1}{2}}}) S_{t+\frac{1}{2}} (Q_{V_{t+\frac{1}{2}}} R_{V_{t+\frac{1}{2}}})^\top \\
&= Q_{U_{t+\frac{1}{2}}} (R_{U_{t+\frac{1}{2}}} S_{t+\frac{1}{2}} R_{V_{t+\frac{1}{2}}}^\top) Q_{V_{t+\frac{1}{2}}}^\top \\
&= Q_{U_{t+\frac{1}{2}}} (P_U P_S P_V^\top) Q_{V_{t+\frac{1}{2}}}^\top \\
&= (Q_{U_{t+\frac{1}{2}}} P_U) P_S (Q_{V_{t+\frac{1}{2}}} P_V)^\top
\end{aligned} \tag{12}$$

Since the product of two orthogonal matrices is orthogonal, $U_{t+1} = Q_{U_{t+\frac{1}{2}}} P_U$ and $V_{t+1} = Q_{V_{t+\frac{1}{2}}} P_V$ retain column orthogonality. Furthermore, $S_{t+1} = P_S$ is diagonal by definition. Thus, we have:

$$\Delta W_{t+1} = U_{t+\frac{1}{2}} S_{t+\frac{1}{2}} V_{t+\frac{1}{2}}^\top = U_{t+1} S_{t+1} V_{t+1}^\top \tag{13}$$

which completes the proof. □

D.2 Proof of Lemma 2

Proof. Let $U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r]$. For any vector $\mathbf{x}$, the projection of $\mathbf{x}$ onto the subspace spanned by U is given by $UU^\top \mathbf{x}$. Consequently, the residual vector $\mathbf{x} - UU^\top \mathbf{x}$ lies in the orthogonal complement of $\mathrm{Span}(U)$. Letting $f(U) = [\mathbf{x}_1, \ldots, \mathbf{x}_r]$, the term in $\sigma(U)$ is orthogonal to U. Therefore, $\sigma(U)^\top \cdot U = 0$. By symmetry, $\sigma(V)^\top \cdot V = 0$, completing the proof. □

D.3 Proof of Lemma 3

Proof. First consider the special case in which the old nonlinear bases lie in the corresponding new spans, so that the following exact relationships hold:

$$\begin{aligned}
\sigma(U_{t+\frac{1}{2}}) &= \sigma(U_{t+1}) \cdot F_U \\
\sigma(V_{t+\frac{1}{2}}) &= \sigma(V_{t+1}) \cdot F_V
\end{aligned} \tag{14}$$

where F_U and F_V are transition matrices. Substituting them into the nonlinear term yields:

$$\begin{aligned}&\sigma(U_{t+\frac{1}{2}}) \cdot D_{t+\frac{1}{2}} \cdot \sigma(V_{t+\frac{1}{2}})^\top \\ =&(\sigma(U_{t+1})F_U)D_{t+\frac{1}{2}}(\sigma(V_{t+1})F_V)^\top \\ =&\sigma(U_{t+1})(F_U D_{t+\frac{1}{2}} F_V^\top)\sigma(V_{t+1})^\top \end{aligned} \tag{15}$$

From Eq. 14, the transition matrices can be solved as $F_U = \sigma(U_{t+1})^+\sigma(U_{t+\frac{1}{2}})$ and $F_V = \sigma(V_{t+1})^+\sigma(V_{t+\frac{1}{2}})$. By setting $D_{t+1} = F_U D_{t+\frac{1}{2}} F_V^\top$ in Eq.9, the nonlinear matrix is preserved exactly in this special case.

In the general case, define $A = \sigma(U_{t+1})$, $B = \sigma(V_{t+1})$, and $N = \sigma(U_{t+\frac{1}{2}})D_{t+\frac{1}{2}}\sigma(V_{t+\frac{1}{2}})^\top$. Eq.9 gives $D_{t+1} = A^+ N (B^+)^\top$, and hence

$$AD_{t+1}B^\top = P_A N P_B, \qquad P_A = AA^+, \qquad P_B = BB^+. \tag{16}$$

This is the Frobenius-orthogonal projection of N onto the linear space $\{AXB^\top : X \in \mathbb{R}^{r\times r}\}$. It therefore minimizes $\|N - AXB^\top\|_F$, and preservation is exact if and only if $P_A N P_B = N$. □

D.4 Proof of Lemma 4

Before proving Lemma 4, we first introduce a theorem.

Theorem 1 (Golub et al., 1987): Let $\Delta\hat{W}^*_{[r]}$ be the rank-r approximation constructed using the top-r singular values of S^* and the corresponding singular vectors. Specifically, $\Delta\hat{W}^*_{[r]} = U^*[:, 1:r]S^*[1:r, 1:r]V^*[:, 1:r]^\top$. The approximation error is characterized by:

$$||\Delta W^* - \Delta\hat{W}^*_{[r]}||_F^2 = \sum_{i=(r+1):r_{\max}} s_i^2 \tag{17}$$

where $U^*[:, 1:r]$ and $V^*[:, 1:r]$ denote the submatrices containing the top-r columns of U^* and V^*, respectively, and $S^*[1:r, 1:r]$ denotes the principal $r \times r$ submatrix of S^*.

More generally, for any integer $1 \le \ell \le r_{\max}$,

$$\min_{\mathrm{rank}(X)\le\ell} \|\Delta W^* - X\|_F^2 = \|\Delta W^* - \Delta W^*_{[\ell]}\|_F^2 = \sum_{i=\ell+1}^{r_{\max}} s_i^2, \tag{18}$$

where $\Delta W^*_{[\ell]}$ denotes a truncated SVD approximation retaining the leading ℓ components.

Proof. Since $\mathrm{rank}(USV^\top) \le r$, Theorem 1 gives the lower bound. Choosing U, V as the leading singular-vector factors and $S = \mathrm{diag}(s_1, \ldots, s_r)$ attains it. The same optimum is representable by LoRA using $B = US$ and $A = V^\top$. □

D.5 Proof of Lemma 5

Proof. Let $r_{\max} = \min(d, k)$ and $m = \min(2r, r_{\max})$. Consider an admissible OReA+ update $\widehat{\Delta W} = USV^\top + \sigma(U)D\sigma(V)^\top$, where $U^\top U = V^\top V = I$, S is diagonal, and $D \in \mathbb{R}^{r\times r}$. For this representation analysis, zero residual columns in the definition of $\sigma(\cdot)$ are defined to remain zero under column-wise normalization. Denote the optimal approximation error over these admissible factors by $E_+^* = \inf_{U,S,V,D} \left\|\Delta W^* - USV^\top - \sigma(U)D\sigma(V)^\top\right\|_F^2$.

Lower bound. Each branch has rank at most r. Therefore,

$$\begin{aligned}\mathrm{rank}(\widehat{\Delta W}) &\le \mathrm{rank}(USV^\top) + \mathrm{rank}\big(\sigma(U)D\sigma(V)^\top\big) \\ &\le 2r. \end{aligned} \tag{19}$$

Since $\widehat{\Delta W} \in \mathbb{R}^{d\times k}$, its rank is also at most $r_{\max}$, and hence at most m. By the best rank-m approximation result in Theorem 1, every admissible update satisfies $\left\|\Delta W^* - \widehat{\Delta W}\right\|_F^2 \ge \sum_{i=m+1}^{r_{\max}} s_i^2$.

Taking the infimum over the admissible factors gives $E_+^* \geq \sum_{i=m+1}^{r_{\max}} s_i^2$. An empty sum is understood to be zero.

Upper bound. Setting $D = 0$ recovers the linear OReA representation. Let $\Delta W_{[r]}^* = U_r^* S_r^* (V_r^*)^\top$ be a truncated SVD formed from the leading r singular components of ΔW^*. Choosing $U = U_r^*$, $S = S_r^*$, $V = V_r^*$, and $D = 0$ is admissible and yields

$$\left\| \Delta W^* - \Delta W_{[r]}^* \right\|_F^2 = \sum_{i=r+1}^{r_{\max}} s_i^2. \tag{20}$$

Consequently, $E_+^* \leq \sum_{i=r+1}^{r_{\max}} s_i^2$. Combining the two inequalities proves

$$\sum_{i=m+1}^{r_{\max}} s_i^2 \leq E_+^* \leq \sum_{i=r+1}^{r_{\max}} s_i^2. \tag{21}$$

Attainability of the lower bound. The lower bound is attained by an admissible update if and only if that update is a best rank-at-most-m approximation of ΔW^*. The following provides a sufficient condition.

Choose nested truncated SVD approximations $\Delta W_{[r]}^*$ and $\Delta W_{[m]}^*$, and define

$$R = \Delta W_{[m]}^* - \Delta W_{[r]}^*, \qquad C_U = \sigma(U_r^*), \qquad C_V = \sigma(V_r^*). \tag{22}$$

Let $P_U = C_U C_U^+, P_V = C_V C_V^+$ be the orthogonal projectors onto the column spaces of C_U and C_V, respectively, where $(\cdot)^+$ denotes the Moore–Penrose pseudoinverse. If$P_U R P_V = R$,then choosing $D = C_U^+ R (C_V^+)^\top$ gives $C_U D C_V^\top = P_U R P_V = R$. Together with the linear branch $\Delta W_{[r]}^*$, this constructs the update $\widehat{\Delta W} = \Delta W_{[r]}^* + R = \Delta W_{[m]}^*$, which attains the lower bound.

This sufficient condition is not guaranteed for an arbitrary target or a fixed nonlinear mapping. In particular, Lemma 2 establishes $U^\top \sigma(U) = 0$ and $V^\top \sigma(V) = 0$, but it does not imply that the nonlinear features have full column rank or mutually orthonormal columns. Moreover, the update has the structured form

$$\widehat{\Delta W} = [U \mid \sigma(U)] \begin{bmatrix} S & 0 \\ 0 & D \end{bmatrix} [V \mid \sigma(V)]^\top. \tag{23}$$

The core contains no cross-branch blocks, and the nonlinear features are determined by the linear factors. Thus, alignment of the combined left and right spans with the leading singular subspaces alone is insufficient to guarantee attainment of the lower bound.

The result characterizes the approximation capacity of the admissible representation family. It does not guarantee that a particular optimizer or the two-phase training schedule attains either bound. □

E Dataset Details

(1) Natural Language Understanding: We employ the GLUE (General Language Understanding Evaluation) benchmark (Wang et al., 2018), a widely adopted multi-task suite designed to evaluate general-purpose natural language understanding models. In our experiments, we include four representative GLUE tasks: MRPC, CoLA, RTE, and STS-B. These tasks span linguistic acceptability judgment (CoLA), paraphrase identification (MRPC), textual entailment (RTE), and semantic similarity regression (STS-B). Evaluation metrics follow standard practice: Matthews Correlation Coefficient for CoLA, Pearson/Spearman correlation for STS-B, and accuracy for classification tasks such as MRPC and RTE. Detailed task descriptions and dataset statistics are summarized in Table 4.

(2) Commonsense Reasoning: We use a set of commonly used datasets, including BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2019), SocialIQA (Sap et al., 2019), ARC-Easy (Clark et al., 2018), ARC-Challenge (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018). These datasets cover a broad range of commonsense reasoning abilities: BoolQ evaluates binary question answering

Table 4: Task descriptions and dataset statistics for the GLUE benchmark.

Corpus	Task	Metrics	# Labels	# Train	# Val	# Test	Domain
Single-Sentence Tasks							
CoLA	Acceptability	Matthews Corr.	2	8.55k	1.04k	1.06k	misc.
Similarity and Paraphrase Tasks							
MRPC	Paraphrase	Accuracy	2	3.67k	408	1.73k	News
STS-B	Sentence similarity	Pearson	1	5.75k	1.5k	1.38k	misc.
Inference Tasks							
RTE	NLI	Accuracy	2	2.49k	277	3k	News & Wikipedia

Table 5: Details of datasets evaluated in the commonsense reasoning task.

Dataset	#Train	#Test	Answer
BoolQ	9.4k	3270	Yes/No
PIQA	16.1k	1830	Option
SIQA	33.4k	1954	Option
ARC-e	1.1k	2376	Option
ARC-c	2.3k	1172	Option
OBQA	5.0k	500	Option

grounded in short passages, PIQA focuses on physical commonsense and action plausibility, and SocialIQA measures the ability to infer social intentions and interpersonal dynamics. ARC-Easy and ARC-Challenge test scientific reasoning under different difficulty levels, requiring multi-step logical inference, while OpenBookQA assesses whether models can combine core scientific facts with external commonsense knowledge. More details are provided in Table 5.

F Parameters Setting Details

F.1 Two-Phase Training Schedule for OReA+

As described in Section 3.4, OReA+ is trained in two sequential phases controlled by a single hyperparameter, the freeze epoch T_1. **Phase 1 (epochs** 0 **to** T_1**):** the nonlinear core D is held at zero (its gradients are accumulated but its weights are reset to zero after every optimizer step) and only the linear factors U, S, V are updated under the QR+SVD retraction of Algorithm 1. **Phase 2 (epochs** T_1 **to** T**):** the linear factors are frozen (`requires_grad=False` on U, S, V) and the optimizer updates only D. The schedule is implemented as a post-optimizer hook on the trainer and applies uniformly to GLUE and commonsense-reasoning runs; the freeze epoch T_1 is task-specific (see the per-track descriptions below and the released configs). For the standard OReA configuration (no nonlinear branch), the entire training corresponds to Phase 1 only.

F.2 Hyperparameter Settings for Natural Language Understanding

We evaluate our proposed OReA on the GLUE benchmark with RoBERTa-Base[1]. All models are optimized using AdamW with a learning rate of 5×10^{-5}, employing a linear scheduler with a warmup ratio of 0.06. To ensure training stability, we set the gradient accumulation steps to 1000. We adjust the training duration based on dataset size: CoLA is trained for 120 epochs, while STS-B, RTE, and MRPC are trained for 80 epochs. These experiments were conducted on four NVIDIA A100 80GB PCIe GPUs. All results are reported as the average over 5 independent runs.

[1] https://huggingface.co/FacebookAI/roberta-base

Table 6: Hyperparameter setup of OReA on GLUE benchmark.

Hyperparameter		STS-B		RTE		MRPC		CoLA	
		$r=8$	$r=2$	$r=8$	$r=2$	$r=8$	$r=2$	$r=8$	$r=2$
Common	Optimizer	AdamW							
	LR Schedule	Linear							
	Learning Rate	5×10^{-5}							
	Warmup Ratio	0.06							
	Max Seq. Len	512							
	Seed	[42, 12345, 56789, 888, 8888]							
	Grad. Accum. Steps	1000							
Specific	α	16	4	16	4	16	4	16	4
	Epochs	80	80	80	80	80	80	120	120
	Batch Size	16	16	16	16	64	64	32	32

Table 7: The performance of different fine-tuning methods on four datasets of the GLUE benchmark, using the RoBERTa-Base model. The reported results are the average of five independent runs. For all datasets, a higher value is better.

Rank r	**Method**	**MRPC**	**STSB**	**RTE**	**COLA**
-	Full Fine-Tuning	89.56	91.20	78.70	63.60
$r=2$	LoRA	88.58	90.74	77.62	62.03
	LoRA+	88.59	90.76	77.65	62.27
	PiSSA	88.78	90.59	77.64	63.08
	LoRAN	88.67	90.45	78.48	63.26
	CLoRA	88.82	90.62	77.88	62.64
	StelLA	88.95	**90.78**	**78.53**	**63.42**
	ours(OReA)	89.15	90.53	78.29	62.21
	ours(OReA+)	**89.21**	90.58	78.33	62.32
$r=8$	LoRA	88.04	90.66	77.55	63.35
	LoRA+	88.62	90.65	77.98	62.58
	PiSSA	88.28	90.57	78.04	63.34
	LoRAN	88.54	90.66	**78.51**	63.49
	CLoRA	87.95	90.68	77.58	62.92
	StelLA	88.69	90.72	78.50	63.57
	ours(OReA)	**88.97**	90.86	78.01	**64.39**
	ours(OReA+)	88.96	**90.87**	78.12	64.35

F.3 Hyperparameter Settings for Commonsense Reasoning

For commonsense reasoning tasks, we employ OReA or OReA+ with a rank $r = 8$, a scaling factor $\alpha = 16$. LLaMA3.1-8B[2], and Qwen2.5-7B[3] are fine-tuned for 8 epochs using a learning rate of 1×10^{-5} with a 100-step linear warmup, a batch size of 4, and a gradient accumulation of 2 steps. All experiments are conducted in BFloat16 precision on a single NVIDIA H100 GPU, targeting the projection matrices $\{W_q, W_k, W_v, W_{up}, W_{down}\}$. When implementing OReA+, we adopt a phased optimization strategy: during the first 4 epochs, only the U, S, V components are updated, with QR+SVD decomposition performed every 100 steps; in the subsequent 4 epochs, U, S, V are frozen, and optimization is restricted to the residual matrix D. For the standard OReA configuration, the training is limited to 4 epochs, following the same optimization procedure as the first stage of OReA+. All results are reported as the average over 5 independent runs.

[2]https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[3]https://modelscope.cn/models/Qwen/Qwen2.5-7B-Instruct

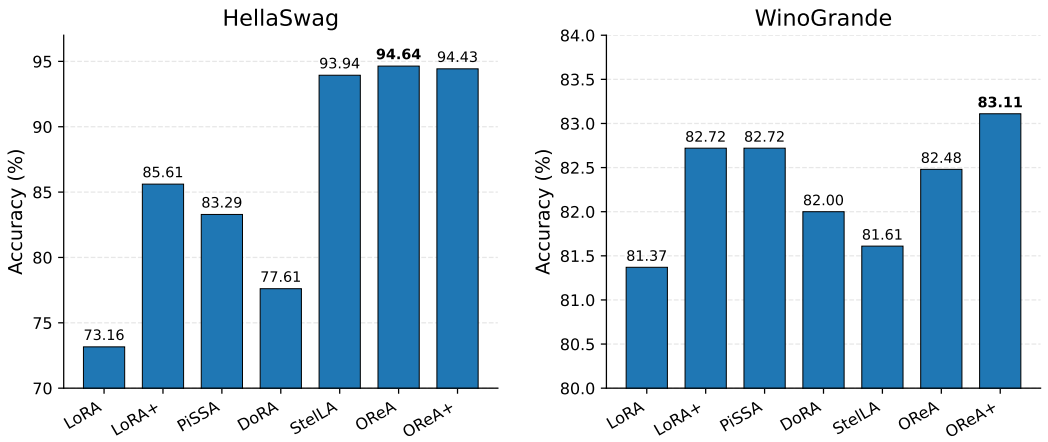

Figure 4: Additional LLaMA3.1-8B results on HellaSwag and WinoGrande.

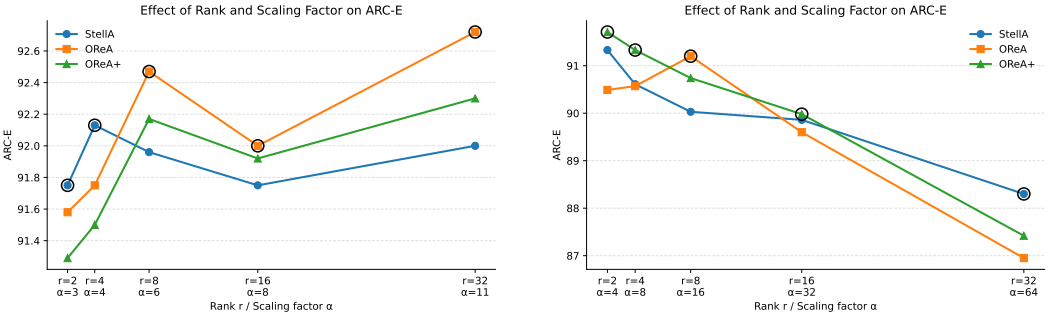

Figure 5: Effect of rank r and scaling factor α on ARC-E.

G Additional Experiments

G.1 Experiment in Natural Language Understanding Task

OReA+ demonstrates competitive NLU performance against full Fine-Tuning method. In Table 7, we compare the proposed OReA and OReA+ with full Fine-Tuning and state-of-the-art PEFT methods on the GLUE benchmark using RoBERTa-Base. Our methods achieve competitive results across tasks and ranks, often achieving similar performance with full Fine-Tuning and other PEFT methods. Since different PEFT methods have similar performance with the full Fine-Tuning method, the advantage of OReA and OReA+ cannot be revealed in the NLU task.

G.2 Additional commonsense benchmarks.

Figure 4 extends the LLaMA3.1-8B evaluation to HellaSwag (Zellers et al., 2019) and WinoGrande (Sakaguchi et al., 2021). OReA records 94.64 on HellaSwag, compared with 93.94 for StelLA, while OReA+ records 83.11 on WinoGrande, compared with 81.61 for StelLA. These observations broaden the evaluated task set for this backbone; the extended two-benchmark evaluation is limited to LLaMA3.1-8B.

G.3 Learning-rate sensitivity.

Figure 6 compares five learning rates on ARC-C and ARC-E. OReA or OReA+ achieves the highest score at every tested learning rate on both tasks, although the leading variant changes across settings. The best recorded scores are 79.78 on ARC-C (OReA+) and 92.30 on ARC-E (OReA). These results show strong performance across the tested range.

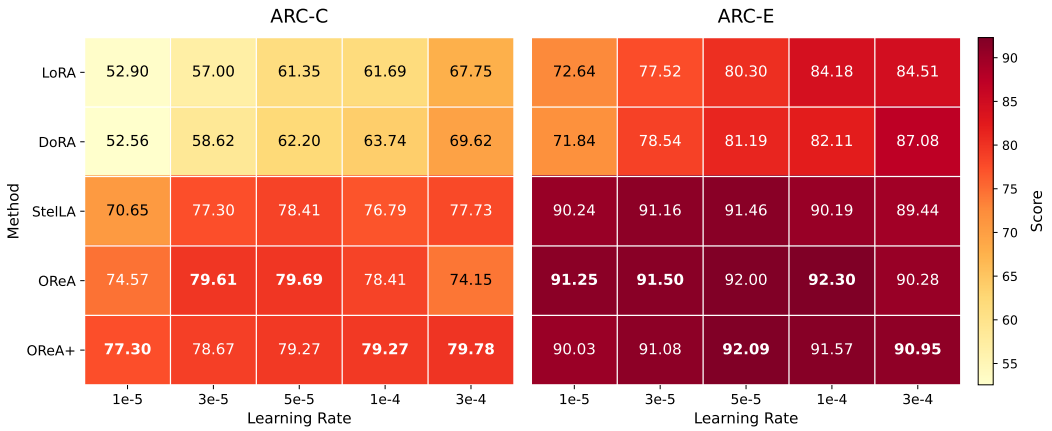

Figure 6: Performance of different methods across learning rates on ARC-C and ARC-E. Bold values indicate the best result at each learning rate.

Table 8: Training time comparison (in minutes). The values in parentheses denote the additional time cost relative to LoRA. We set $epoch = 80$.

Dataset	LoRA	Ours (OReA)
STS-B	101.7	144.2 (0.42×)
RTE	48.5	62.8 (0.29×)
MRPC	70.8	74.9 (0.06×)
CoLA	155.1	179.1 (0.15×)

G.4 Rank–scaling interaction.

On ARC-E (Figure 5), under $\alpha \in \{3, 4, 6, 8, 11\}$, OReA obtains the largest recorded value at ranks 8, 16, and 32, whereas StelLA is stronger at ranks 2 and 4. Under $\alpha = 2r$, the ordering changes and all three methods have lower scores at the largest tested rank.

With the scaling convention $\gamma = \alpha/r$, the second schedule keeps $\gamma = 2$ constant, while the first approximately decreases γ as $2/\sqrt{r}$. These observations should therefore be interpreted as a joint rank–scaling sensitivity study, not as an isolated causal effect of rank. Rank-dependent scaling has also been studied in rsLoRA (Kalajdzievski, 2023).

Given the rank–scaling sensitivity in Figure 5, we further examine whether the advantage over LoRA and DoRA persists at a higher rank under the common $\alpha = 2r$ rule. On ARC-C, at $r = 16$ and $\alpha = 32$, OReA and OReA+ score 74.91 and 76.02, respectively, compared with 65.19 for LoRA and 65.70 for DoRA. The advantage over these two baselines therefore persists in this additional configuration.

G.5 Spectra of the learned adapter updates.

We analyze the final LoRA and OReA+ checkpoints on LLaMA3.1-8B ARC-E at nominal ranks $r \in \{4, 8, 16, 32, 64\}$, with $\alpha = 2r$. To compare the same weights, we average over the 160 shared q, k, v, up, and down projection layers. For each layer, we form the represented update ΔW and define its effective rank as $\exp(-\sum_i p_i \log p_i)$, where $p_i = \sigma_i^2 / \sum_j \sigma_j^2$ and σ_i are the singular values of ΔW. Figure 7 shows that the mean effective ranks of LoRA are 2.76, 4.54, 7.14, 11.28, and 17.30, respectively, while those of OReA+ are 5.57, 10.43, 18.54, 33.39, and 65.77. OReA+ therefore uses more than r effective directions at all five ranks, consistent with its rank-at-most-$2r$ update structure. The fraction of OReA+ update energy beyond the leading r singular directions is 21.3%, 19.2%, 10.9%, 2.6%, and 2.6%, respectively: the complementary directions carry appreciable energy at small ranks, but their relative contribution decreases as r grows.

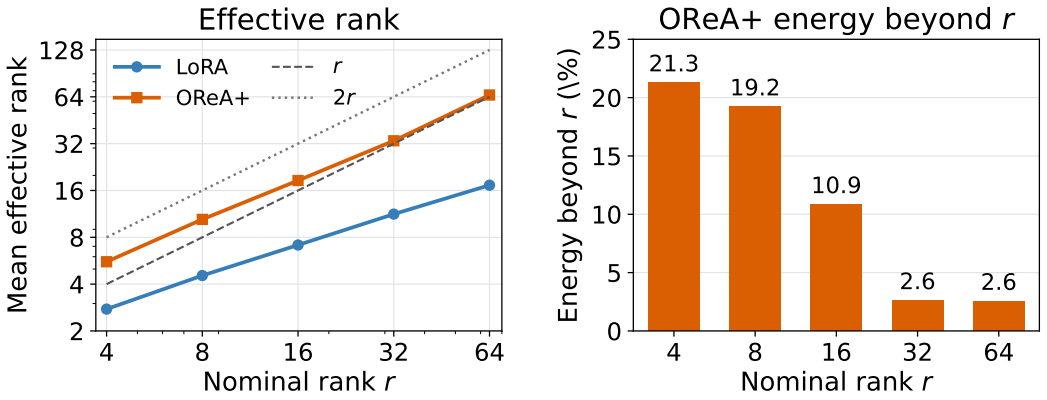

Figure 7: Final adapter-update spectra on LLaMA3.1-8B ARC-E, averaged over the 160 projection layers shared by LoRA and OReA+. Left: mean effective rank versus nominal rank (both axes use logarithmic scales); dashed and dotted lines indicate r and $2r$. Right: fraction of OReA+ squared-singular-value energy beyond the first r directions.

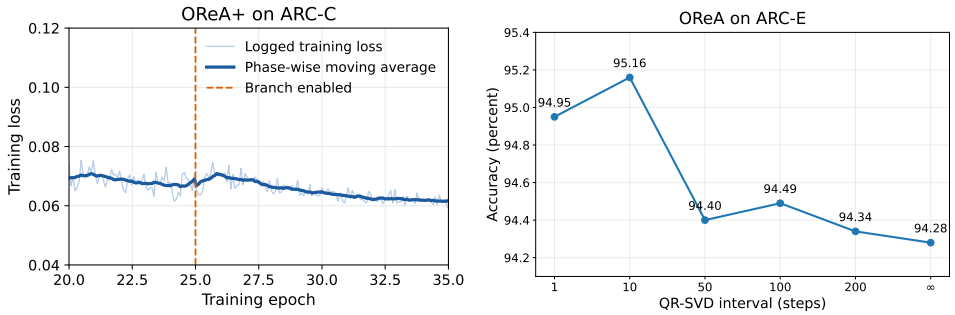

Figure 8: OReA+ component ablations. Left: Qwen2.5-7B ARC-C training loss over epochs 20–35, with raw logged values in pale blue and phase-wise moving averages in dark blue; the nonlinear branch is enabled at epoch 25. Right: Qwen2.5-7B ARC-E accuracy at QR-SVD re-factorization intervals $q \in \{1, 10, 50, 100, 200, \infty\}$ with the nonlinear branch disabled ($r = 8$, $\alpha = 16$).

G.6 Nonlinear-branch and re-factorization-interval ablations.

We examine the staged nonlinear extension and the frequency of orthogonality restoration in the OReA+. Figure 8 (left) presents an extended-training diagnostic, separate from the 4+4-epoch schedule used for Table 1. We first train OReA for 25 epochs, by which point the training loss has plateaued under the tested configuration. We then freeze U, S, V and optimize only D. The OReA-stage loss fluctuates around 0.07 before epoch 25; after the nonlinear branch is enabled, the loss trends downward, reaching 0.0595 near epoch 35. It indicates that the complementary branch can further improve training fit after the observed linear-training plateau. The right plot shows the five-point OReA re-factorization-interval sweep on Qwen2.5-7B ARC-E with the nonlinear branch disabled. Accuracy peaks at 95.16% for $q = 10$, compared with 94.95% for $q = 1$, and is generally lower for $q \in \{50, 100, 200\}$ (94.32–94.49%). The small reduction at $q = 1$ may reflect floating-point roundoff accumulating across overly frequent QR-SVD re-factorizations; increasing the interval to $q = 10$ avoids some of these repeated operations, whereas still larger intervals restore the structure less often. Thus, a moderate restoration interval performs best among those tested, and training fit improves further after the nonlinear branch is activated. We further include $q = \infty$, which disables QR–SVD re-factorization throughout training while keeping all other settings unchanged. Its lower accuracy compared with the finite-q settings provides direct evidence for the benefit of periodic re-factorization.

G.7 Training Time Analysis

We compare the training time between our method and the standard LoRA. The main time cost is the operation for orthogonality constraints. We compare OReA with LoRA to carefully compare the difference of orthogonality constraints in Table 8. It shows the additional cost is minimal, demonstrating the efficiency of our method.