Extended-SVD Low-Rank Adaptation: PLORA for Parameter-Efficient Fine-Tuning

Ratan Kumar Gulipilli
Indian Institute of Technology Patna
NTPC Limited, India
ratan_24a03res74@iitp.ac.in

Abstract

We introduce Extended-SVD Low-Rank Adaptation, a parameter-efficient framework that extends the pretrained singular subspace while retaining a frozen spectral coordinate system and fixed singular-value-derived scaling. Its stabilized variant, PLORA (Procrustes Low-Rank Adaptation), applies localized Procrustes-inspired reorientation to the extended basis and confines task-specific learning to compact routing matrices P, R, and Q.

Across GLUE and vision benchmarks, PLORA achieves competitive performance with substantially fewer trainable parameters. In the full-network comparison, PLORA-r8 uses substantially fewer trainable parameters, although the performance gap varies by task. Beyond accuracy, PLORA produces task- and architecture-dependent changes in conditioning, effective rank, and subspace evolution.

Ablations show that Random Bases produce poorer routing-core conditioning and markedly lower top-r right-subspace overlap in the evaluated NLP tasks. In the evaluated settings, pretrained spectral bases are associated with more favorable optimization geometry, while Procrustes reorientation acts as an optimization-geometric regularizer. Overall, Extended-SVD provides a structured low-rank adaptation framework for studying the role of pretrained geometry in PEFT.

1 Introduction

Large language models exhibit highly structured pretrained weight spaces where dominant singular directions encode substantial task-relevant information. Parameter-efficient fine-tuning (PEFT) methods like LoRA (Hu et al., 2022) exploit this through low-rank updates while freezing the backbone. Recent SVD-based approaches further incorporate pretrained spectral geometry into adapter initialization.

Existing methods typically partition the singular spectrum, adapting either dominant (Meng et al., 2024) or lower-energy (Wang et al., 2025) regions. In contrast, our approach treats nearby lower-energy vectors as a geometric continuation of the principal pretrained subspace.

We first define the Extended-SVD architecture, which retains an extended frozen singular basis and performs adaptation through a compact trainable routing core (referred to experimentally as the Unaligned configuration). To mitigate observed conditioning instability within this expanded space, we introduce PLORA (Procrustes Low-Rank Adaptation). PLORA applies Procrustes-inspired coordinate reorientation to the extended bases, establishing this Reoriented (Aligned) configuration as the default stabilized setup.

Our contributions are:

- We introduce the Extended-SVD architecture, a PEFT framework based on a structured PRQ decomposition that separates frozen pretrained geometry from compact trainable routing dynamics.
- We propose PLORA (Procrustes Low-Rank Adaptation), which applies Procrustes-inspired coordinate reorientation to structure these extended bases and stabilize optimization.
- We provide empirical analysis across NLP and vision benchmarks showing that alignment substantially reduces the observed routing-core condition-number proxy in the evaluated ViT configuration, while NLP effects are task-dependent.
- Across GLUE and vision benchmarks, PLORA achieves competitive performance under substantially reduced parameter budgets. In particular, PLORA-r8 achieves competitive performance against the standard attention-only LoRA baseline on GLUE while using substantially fewer trainable parameters and

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 74–93
November 6-10, 2026 ©2026 Association for Computational Linguistics

PLORA-$r16$ achieves strong overall performance while retaining comparable peak task performance in the evaluated settings.

Rather than positioning Extended-SVD as a universal replacement for LoRA, we present it as a controlled framework for studying how geometric constraints influence optimization dynamics, reproducibility, and parameter efficiency in modern PEFT systems.

2 Related Work

Parameter-Efficient Fine-Tuning. Beyond full fine-tuning, PEFT methods mitigate the scale-related costs of LLMs (Han et al., 2024). These strategies generally fall into three categories: *structural adapters* that insert trainable modules between frozen layers (Houlsby et al., 2019; Pfeiffer et al., 2021; Rücklé et al., 2021; Hu et al., 2023); *prompt-based tuning* that prepends learnable tokens (Lester et al., 2021; Li and Liang, 2021; Liu et al., 2022; Zhang et al., 2025); and *sparse updates* that restrict learning to a subset of existing weights via masks or diff-pruning (Guo et al., 2021; Sung et al., 2021; Ansell et al., 2022). Recent work explores unified frameworks to automate these paradigms (Mao et al., 2022; Zhou et al., 2024; Chen et al., 2023; Lawton et al., 2023).

Low-Rank Adaptation. LoRA (Hu et al., 2022) is a dominant PEFT paradigm motivated by the low effective dimensionality of neural network parameters (Denil et al., 2013) and prior demonstrations of low-rank weight factorization (Zhang et al., 2014). Extensions improve efficiency through dynamic ranks (Zhang et al., 2023; Mao et al., 2024), search-free adaptation (Valipour et al., 2023), quantization (Dettmers et al., 2023), gradient projection (Zhao et al., 2024), memory-efficient optimization (Luo et al., 2024), importance sampling (Pan et al., 2024; Li et al., 2025), pruning (Zhou et al., 2025), and weight tying (Renduchintala et al., 2024; Liu et al., 2023).

Expressivity and Geometry. To break LoRA's linear constraints, methods like AuroRA (Dong et al., 2025), LoRAN (Li et al., 2024), and PEANuT (Zhong et al., 2026) inject non-linear activations, while Mixture-of-Experts variants route inputs dynamically (Feng et al., 2024; Wu et al., 2024; Liu et al., 2024). More relevant to our work, SVD-based adaptation improves initialization using pretrained spectral geometry: PiSSA (Meng et al., 2024) utilizes dominant singular vectors, while MiLoRA (Wang et al., 2025) targets lower-energy components. Other geometric or frequency-domain approaches include DFT-based optimization (Gao et al., 2024), geometric integration (Schotthöfer et al., 2025), and sine-activated matrices (Ji et al., 2025).

Unlike these methods, the Extended-SVD framework explicitly factorizes adaptation into frozen structured bases and compact trainable routing matrices. This contrasts with frozen-basis methods like VeRA (Kopiczko et al., 2024), which uses random projections primarily for memory reduction. By leveraging pretrained singular subspaces to construct extended adaptation bases, the Extended-SVD architecture provides a controlled framework to analyze conditioning and subspace evolution. We evaluate the framework across NLP and vision architectures, with PLORA providing the re-oriented configuration studied in the experiments.

3 Method

3.1 Preliminaries: Standard Low-Rank Adaptation

Standard LoRA modifies a pretrained weight matrix $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ by adding a low-rank update $\Delta W = AB$, where $A \in \mathbb{R}^{d_{\text{out}} \times r}$ and $B \in \mathbb{R}^{r \times d_{\text{in}}}$, with $r \ll \min(d_{\text{out}}, d_{\text{in}})$. The forward pass becomes $h = W_0 x + \alpha(AB)x$, where α is a scaling factor. During training, W_0 remains frozen while A and B are optimized.

3.2 Extended-SVD Adaptation

The Extended-SVD framework reformulates the unscaled weight update as a structured five-matrix cascade (Figure 1):

$$\Delta W_{\text{raw}} = U_{\text{base}} P R Q V_{\text{base}}^T \tag{1}$$

The applied update is then scaled by a constant factor α, defined as $\Delta W_{\text{app}} = \alpha \Delta W_{\text{raw}}$, yielding the final forward pass $W_t = W_0 + \Delta W_{\text{app}}$.

Here,

- $U_c \in \mathbb{R}^{d_{\text{out}} \times r_{\text{ext}}}$ and $V_c \in \mathbb{R}^{d_{\text{in}} \times r_{\text{ext}}}$ are frozen SVD-derived structural matrices obtained after the Procrustes-inspired block transformation.
- $D \in \mathbb{R}^{r_{\text{ext}} \times r_{\text{ext}}}$ is a fixed diagonal scaling matrix containing the square roots of the extended singular values,
- $U_{\text{base}} = U_c D$ and $V_{\text{base}} = V_c D$ are the frozen structural factors,

- $r_{\text{ext}} = r \cdot m$ is the extended subspace dimension,
- $m \geq 1$ is the extension multiplier,
- $P \in \mathbb{R}^{r_{\text{ext}} \times r}$ and $Q \in \mathbb{R}^{r \times r_{\text{ext}}}$ are trainable routing matrices,
- $R \in \mathbb{R}^{r \times r}$ is the trainable core mixing matrix.

Only P, R, and Q are trainable; U_c, V_c, and D remain strictly frozen throughout optimization.

Procrustes-Inspired Coordinate Reorientation. Let U_{top} and U_{mid} denote the principal and contiguous extension blocks from the pretrained SVD, and define the corresponding blocks for V analogously. Their cross-overlap matrix $M = U_{\text{top}}^{\top} U_{\text{mid}}$ is zero in exact arithmetic, although its finite-precision realization may contain small numerical residuals.

In our implementation, the computed overlap matrix is factorized using a thin SVD, $M = L\Sigma R^{\top}$. Let $k_{\min} = \min(k_{\text{top}}, k_{\text{mid}})$. The implementation constructs the square block $H = L_{1:k_{\min},:} R^{\top}_{:,1:k_{\min}}$, embeds it into the leading $k_{\min} \times k_{\min}$ block of an identity matrix, and right-multiplies the extension block by the resulting transformation. This operation is applied separately to the left and right structural blocks.

We refer to this operation as a Procrustes-inspired, SVD-derived coordinate transformation, rather than an exact orthogonal Procrustes alignment. In the rectangular-overlap case, the selected square block H is not guaranteed to be orthogonal. Consequently, the implementation does not guarantee that the transformed extension columns remain orthonormal or that the transformed extension block retains exactly the same span as its original counterpart.

The transformed blocks are concatenated with the principal blocks to form the frozen structural matrices U_c and V_c. The fixed diagonal scaling D, derived from the original singular values, is then applied to obtain $U_{\text{base}} = U_c D$ and $V_{\text{base}} = V_c D$. The construction leaves the allocated extension-block width and trainable parameter count unchanged. Its effects on optimization geometry and task performance are therefore evaluated empirically rather than attributed to a guaranteed orthogonal transformation. Because the cross-overlap may be dominated by floating-point residuals, the resulting transformation may also depend on numerical precision and the linear-algebra implementation.

The number of trainable parameters is

$$N_{\text{PLORA}} = r_{\text{ext}} r + r^2 + r r_{\text{ext}} = r^2(2m + 1). \quad (2)$$

Unlike standard LoRA, whose trainable parameters scale directly with layer dimensions, $N_{LoRA} = r(d_{in} + d_{out})$, the Extended-SVD framework decouples the trainable parameter footprint from the width of the target layer, although frozen basis dimensions scale with model dimensionality.

Relation to PiSSA, MiLoRA, and VeRA: To clarify our core contribution, we must distinguish SVD-based initialization from geometry-constrained optimization. PiSSA performs $W = U\Sigma V^T$ and initializes trainable adapters as $A = U_r \Sigma_r^{1/2}$ and $B = \Sigma_r^{1/2} V_r^T$. Optimization proceeds through $W = W_{\text{res}} + AB$. Because A and B are fully trainable, the singular vectors themselves are updated during fine-tuning, meaning the pretrained spectral coordinate system is ultimately lost. MiLoRA similarly utilizes SVD strictly for initialization, whereas VeRA employs frozen random projections that completely discard pretrained geometry.

In contrast, the Extended-SVD framework constructs an extended, strictly frozen coordinate system (U_c and V_c) derived from the pretrained singular vectors, with the reorientation transformation providing a structured extension. Fixed singular-value-derived scaling (D) is then applied to preserve pretrained singular-value-derived scaling. Task adaptation is parameterized exclusively via the unscaled update $\Delta W_{\text{raw}} = U_{\text{base}} P R Q V_{\text{base}}^T$, where $U_{\text{base}} = U_c D$ and $V_{\text{base}} = V_c D$. Our novelty lies in retaining a frozen pretrained-derived coordinate system and fixed singular-value-derived scaling while driving adaptation entirely through the trainable routing dynamics (P, R, Q). Consequently, optimization remains strictly confined within the pretrained coordinate system.

Scaling conventions across ablations. The primary Aligned and Unaligned Extended-SVD configurations and the structural ablations Random-Bases, Random-Tail, and No-PQ apply the learned update as $\Delta W_{\text{app}} = \alpha \Delta W_{\text{raw}}$. PiSSA and MiLoRA use their native AB updates without an additional scaling multiplier, while VeRA retains its native module scaling. Unless otherwise stated, $\alpha = 16$.

3.3 Initialization and Optimization Geometry

Extended-SVD initializes frozen bases from a thin SVD, $W_0 = U\Sigma V^T$, extracting the leading r_{ext}

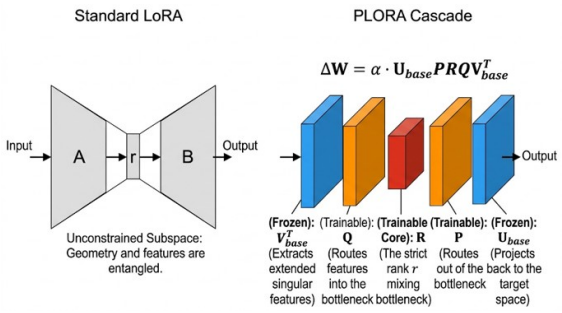


Figure 1: Comparison between standard LoRA (left) and the Extended-SVD architecture (right). Standard LoRA learns unconstrained low-rank projections directly, whereas Extended-SVD fixes an extended pretrained coordinate system through frozen singular bases and restricts adaptation to the compact routing cascade PRQ.

components to expand the top-r principal subspace to $r_{\text{ext}} = r \cdot m$. PLORA applies localized Procrustes reorientation to this extended coordinate system, while the Unaligned variant retains the original ordering.

Matrices P and Q are randomly initialized, whereas $R \leftarrow 0$, giving $\Delta W_{\text{raw}} = 0$ initially. During training, P and Q provide extended routing flexibility and R performs compact mixing. The reorientation is therefore treated as a geometry-aware optimization regularizer. At inference, the trained cascade is merged into $W_{\text{inf}} = W_0 + \alpha U_{\text{base}} PRQV_{\text{base}}^T$, introducing no additional inference overhead.

4 Experimental Setup

Configuration Nomenclature. We evaluate five geometric adaptation configurations throughout the paper:

- **Aligned (PLORA):** Pretrained SVD-initialized structural bases with localized Procrustes reorientation, utilizing extended routing matrices (P, Q). This serves as the default stabilized configuration.
- **Unaligned (Extended-SVD):** Pretrained SVD-initialized structural bases without Procrustes reorientation, utilizing extended routing matrices (P, Q). This represents the base framework without coordinate reorientation.
- **No-PQ:** Aligned SVD structural bases where the extended routing matrices (P, Q) are removed. The applied update is parameterized directly as $\Delta W_{\text{app}} = \alpha U_{\text{base,top}} R V_{\text{base,top}}^T$, restricting the frozen bases strictly to the top-r principal directions ($r_{\text{ext}} = r$). Trainable parameters are reduced to r^2.
- **Random Bases**: Random orthonormal structural directions that replace the pretrained singular-vector directions while retaining the fixed singular-value-derived scaling (D).
- **Random-Tail (R-Tail)**: Aligned SVD structural bases where the contiguous mid-spectrum extension is replaced by randomly selected non-principal singular vectors, evaluating the necessity of spectral continuity.

Internally, these configurations are implemented using method-specific configuration flags; throughout the manuscript we use the descriptive labels above for clarity.

We adopt a staged evaluation protocol under fixed computational budgets. Phase 1 performs controlled GLUE ablations to study the interaction between rank (r), extension multiplier (m), and coordinate reorientation. Phase 2 evaluates the selected configurations on the seven-task GLUE benchmark (Wang et al., 2019) using the RoBERTa-base architecture (Liu et al., 2019) without extensive task-specific retuning. Phase 3 isolates Aligned versus Unaligned behavior on SST-2, CoLA, RTE, and STS-B to analyze optimization geometry.

All primary configurations are evaluated across three random seeds (42, 123, 456) unless otherwise stated, with results reported as mean $\pm$ standard deviation. We additionally evaluate the Extended-SVD framework on the CIFAR-100 dataset (Krizhevsky, 2009) using ResNet-50 (He et al., 2016) and ViT-B/16 (Dosovitskiy et al., 2021) to test whether the observed geometric effects transfer beyond NLP.

For NLP, we fine-tune RoBERTa-base using AdamW, linear warmup, and early stopping. Vision experiments use standard cosine-decay schedules with warmup and gradient clipping. Unless otherwise noted, the same PLORA formulation is used across domains, with only the external training schedule adjusted per task.

Model	Params	Train.	%	Tasks
RoBERTa-base	125M	0.608M	0.49%	seven-task GLUE Benchmark
ViT-B/16	86M	0.178M	0.21%	CIFAR-100
Gemma-2B	≈2.5B	1.06M	≈0.042%*	RTE, COLA, STS-B, MRPC

Table 1: Model configurations and reported trainable parameter budgets for $r = 16, m = 16$. For RoBERTa-base and Gemma-2B, the reported parameters strictly isolate the Extended-SVD adapters, explicitly excluding the classification heads (e.g., the $\sim$0.59 M parameter RoBERTa head) for direct comparability with standard PEFT baselines. Overall, the Gemma-2B configuration adapts approximately 0.04% of the backbone. For ViT-B/16, the total trainable budget includes both 0.101 M adapter parameters and a 0.077 M classification head.*Parameter percentage is computed relative to the approximately 2.5B non-embedding backbone parameters; the model name "2B" conventionally excludes the embedding parameters.

5 Results

5.1 Natural Language Processing: The GLUE Benchmark

Across GLUE, the Extended-SVD framework remains competitive under compressed budgets (Table 2), often approaching full-network LoRA performance at a substantially lower trainable cost (Table 3). Procrustes reorientation primarily alters optimization dynamics rather than consistently improving peak accuracy, with seed variance effects remaining strictly task-dependent. As illustrated in Figure 3, Aligned and Unaligned projections perform comparably on semantic tasks at higher capacities ($r = 16$), indicating that coordinate reorientation enforces localized subspace concentration without sacrificing peak expressivity.

Parameter Counts and Scaling Behavior. In many prior PEFT baselines, standard LoRA is applied only to Query and Value attention matrices. Extending LoRA to all 72 linear layers substantially increases parameters because its complexity scales with layer width, $N_{\text{LoRA}} \propto r(d + k)$. In contrast, the Extended-SVD architecture relies on frozen structural bases and compact routing matrices, yielding $N_{\text{Ext-SVD}} \propto r^2(2m + 1)$. Consequently, the framework adapts all 72 layers using approximately 0.15M trainable parameters at $r = 8$, $m = 16$, whereas full-network LoRA requires approximately 1.3M parameters under the same setting.

Because this parameter count is largely decoupled from layer width, we evaluate the framework under full-network adaptation settings (see Table 1 for total trainable budgets).

5.2 Vision Domain: Extreme Parameter Efficiency

In vision, the Extended-SVD framework shows especially strong behavior under aggressive parameter compression. On ViT-B/16, the Aligned configuration (PLORA) achieves $92.37\% \pm 0.20\%$ on CIFAR-100 using 0.178 M total trainable parameters, reporting a higher point estimate than the cited adaptation baselines while maintaining a substantially smaller parameter footprint (Table 4). On ResNet-50, PLORA also remains competitive relative to established low-rank adaptation baselines. Specifically, PLORA uses $20.5\times$ fewer adapter parameters and $7.4\times$ fewer total trainable parameters than LoRA-C (Table 4).

Crucially, the geometric impact of Procrustes reorientation is highly domain-asymmetric. In structured vision architectures, the optimization landscape can become sharply ill-conditioned when adaptation capacity is tightly constrained. Here, in the evaluated ViT configuration, PLORA acts as an empirical geometric stabilizer, substantially reducing the observed routing-core condition-number proxy. Conversely, as analyzed further in Section 6.1, NLP models exhibit a different task-dependent response to reorientation, with changes in conditioning and effective-rank utilization that are not uniformly aligned with the vision behavior.

To isolate the effect of coordinate reorientation on optimization stability, we track the condition number of the core matrix R during ViT-Base/16 training (Figure 2) at $r = 16, m = 16$. The Unaligned configuration exhibits severe condition-number oscillation and instability (mean $\kappa(R) \approx 1572 \pm 1546$), whereas the Aligned variant remains tightly bounded ($\kappa(R) < 205$) in the evaluated training traces, indicating substantially more stable optimization dynamics under alignment.

5.3 Convergence Analysis

The Extended-SVD framework frequently reaches competitive validation performance under highly

Method	Params	SST-2	MRPC	CoLA	QNLI	RTE	STS-B	MNLI-m/mm	Avg (6)
Full FT*	125M	94.8	90.2	63.6	92.8	78.7	91.2	87.6/86.9	85.2
BitFit*	0.1M	93.7	92.7	62.0	91.8	81.5	90.8	84.7/84.4	85.4
AdapterD*	0.9M	94.7	88.4	62.6	93.0	75.9	90.3	85.1/84.8	84.2
LoRA*	0.3M	95.1	89.7	63.4	93.3	78.4	91.5	86.6/86.0	85.2
AdaLORA*	0.3M	94.5	88.7	62.0	93.1	81.0	90.5	86.8/86.1	85.0
DyLoRA*	0.3M	94.3	89.5	61.1	92.2	78.7	91.1	—	84.5
FourierFT*	0.024M	94.2	90.0	63.8	92.2	79.1	90.8	—	85.0
LoRA-drop*	0.15M	94.5	89.5	62.9	93.1	81.4	91.0	—	85.4
DoRA*	0.3M	95.0	89.7	64.9	92.9	79.2	91.3	—	85.5
AuroRA*	0.075M	95.2	91.9	65.1	93.4	85.2	91.5	—	87.1
PLORA-r4	0.038M	93.9	88.7	61.7	91.0	75.4	90.7	84.5/84.7	83.6
PLORA-r8	0.152M	94.6	89.5	65.1	91.7	76.5	90.9	85.4/85.5	84.7
PLORA-r16	0.608M	95.1	89.5	64.2	92.5	79.1	90.9	87.1/86.7	**85.2**

Table 2: Peak performance comparison of PLORA and baseline PEFT methods on the GLUE benchmark. Baseline metrics for comparison methods, marked with (*), are taken directly from the AuroRA paper (Dong et al., 2025). PLORA results report the peak evaluation performance achieved across three random seeds ($m = 16$). Note that MRPC performance in this table is reported as Accuracy to align with the cited baselines, whereas Table 3 reports F1 scores. Following baseline literature, the average is computed across the first six tasks to maintain consistent task coverage. Published baseline results are shown for contextual comparison; run aggregation protocols differ across methods (e.g., PLORA reports the peak of three seeds, whereas cited baselines largely report five-run medians).

Task	Metric	Rank	LoRA Params	Ext-SVD Params	Standard LoRA	Ext-SVD (Unaligned)	PLORA(Aligned)
SST-2	Accuracy	$r = 8$	∼1.32M	∼0.15M	94.65±0.35	94.30±0.37	94.11±0.46
SST-2	Accuracy	$r = 16$	∼2.65M	∼0.61M	94.46±0.18	94.00±0.35	94.50±0.53
MRPC	F1	$r = 8$	∼1.32M	∼0.15M	92.04±0.29	92.46±0.76	92.15±0.37
MRPC	F1	$r = 16$	∼2.65M	∼0.61M	92.04±0.50	92.24±0.24	91.94±0.34
CoLA	Matthews	$r = 8$	∼1.32M	∼0.15M	63.03±1.14	63.01±2.66	63.12±2.64
CoLA	Matthews	$r = 16$	∼2.65M	∼0.61M	63.29±1.79	62.69±1.93	63.56±0.75
STS-B	Pearson	$r = 8$	∼1.32M	∼0.15M	90.89±0.27	90.70±0.37	90.60±0.28
STS-B	Pearson	$r = 16$	∼2.65M	∼0.61M	90.86±0.12	90.57±0.15	90.70±0.16
RTE	Accuracy	$r = 8$	∼1.32M	∼0.15M	79.66±0.75	73.89±2.56	75.81±0.63
RTE	Accuracy	$r = 16$	∼2.65M	∼0.61M	78.22±1.99	77.86±0.75	77.98±1.08

Table 3: Comparison between standard LoRA and Unaligned Extended-SVD and Aligned (PLORA) for $m = 16$ under matched full-network adaptation settings on representative GLUE tasks. Standard LoRA adapts all 72 linear projections (Q/K/V/O + MLP up/down projections), resulting in substantially larger parameter counts due to dimension-dependent scaling. PLORA achieves competitive performance under substantially smaller parameter budgets.

compressed parameter budgets, but the Aligned (PLORA) and Unaligned configurations often follow different optimization trajectories despite similar final accuracy. These effects are particularly visible on SST-2 and RTE, suggesting that coordinate reorientation primarily influences optimization dynamics within the frozen extended subspace rather than changing model capacity directly.

5.4 Decoder-Only LLM Evaluation

To evaluate whether the PLORA framework extends effectively beyond encoder architectures, we additionally test it on the decoder-only Gemma-2B model across RTE, MRPC, CoLA, and STS-B under a low-parameter adaptation budget (∼1.06M trainable parameters). Assuming an approximate 2.5B total parameter count for the base architecture excluding embeddings, this corresponds to adapting roughly 0.042% of the model. Across tasks, PLORA achieves competitive performance with low variance across seeds, demonstrating stable adaptation on decoder-only architectures. Results are shown in Table 5: RTE 81.95 ± 1.28, MRPC 87.95 ± 0.09, CoLA 61.56 ± 1.10, STS-B 87.94 ± 0.23.

To assess scalability to larger architectures and complex reasoning tasks, we extend PLORA to Llama-3-8B and Qwen2.5-7B. Utilizing 4-bit NF4 quantization, the rank-8 framework adapts all targeted linear projections using just 0.473M and 0.414M trainable parameters, respectively. Both models were fine-tuned for three epochs on Commonsense170K and evaluated on a seven-task subset (BoolQ, PIQA, HellaSwag, WinoGrande, ARC-

Method	Adapter (M)	Trainable (M)	Full (M)	Acc. (%)	Gap (%)	Δ vs. PLORA	Source
ResNet-50							
LoRA-C (Robust)	2.05	2.26	23.80	82.98	—	+1.14	Ding et al. (2024)
PLORA (Aligned)	0.100	0.305	23.81	81.84 ± 0.27	7.03 ± 0.11	—	This work
ViT-B/16							
AdaLoRA	0.45	0.527	85.80	91.44	7.6	−0.93	Schotthöfer et al. (2025)
GeoLoRA (Local)	0.35	0.427	85.70	91.63	—	−0.74	Schotthöfer et al. (2025)
GeoLoRA (Global)	0.47	0.547	85.82	91.62	—	−0.75	Schotthöfer et al. (2025)
Ext-SVD (Unaligned)	0.101	0.178	85.98	92.37 ± 0.09	7.47 ± 0.16	0.00	This work
PLORA (Aligned)	0.101	0.178	85.98	$\mathbf{92.37 \pm 0.20}$	7.52 ± 0.19	—	This work

Table 4: Comparison of parameter efficiency and CIFAR-100 accuracy. PLORA and Ext-SVD configurations on ResNet-50 and ViT-B/16 utilize $r = 16, m = 16$. "Adapter" denotes trainable adapter parameters only, while "Trainable" includes adapters and the classification head. "Gap (%)" denotes the train-test accuracy gap. Baseline metrics are taken directly from their respective cited references.

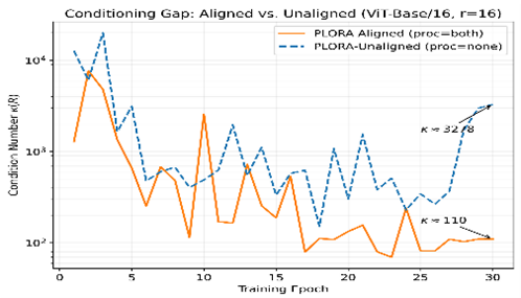


Figure 2: Optimization conditioning on ViT-Base/16 under extreme parameter compression. The plot compares Aligned (PLORA) and Unaligned Extended-SVD configurations at identical capacity ($r = 16, m = 16$); Unaligned training exhibits severe condition-number oscillation, while Aligned training remains below the observed condition-number range of the Unaligned configuration.

Task	Metric	Mean ± Std
RTE	Accuracy	81.95 ± 1.28
MRPC	F1	87.95 ± 0.09
CoLA	Matthews	61.56 ± 1.10
STS-B	Pearson	87.94 ± 0.23

Table 5: PLORA Performance on GLUE Benchmark Semantic Tasks for Gemma-2B (4-bit NF4 quantization) with $r = 16, m = 16$. Results are mean ± standard deviation across three seeds.

Easy/Challenge, OpenBookQA) via the EleutherAI Evaluation Harness. The models were evaluated with 0-shot prompting (no in-context examples) using log-likelihood multiple-choice scoring. While these specific metrics differ from standard generative leaderboards, they demonstrate computational applicability at the 7–8B scale. Full dataset splits, optimization hyperparameters, and single-A100 hardware telemetry are detailed in Appendix A.3.

In addition to task performance, we report the measured execution profile of PLORA on 7–8B models. Because the extended spectral bases remain completely frozen, they require no weight-gradient computation or optimizer updates, despite passing activation gradients. Hardware telemetry on a single NVIDIA A100 GPU (sequence length 128, effective batch 128) provides the measured execution profile for 7B+ models: Llama-3-8B required only 17.54 GiB peak allocated VRAM, achieving 9.01 samples/s throughput and 215.76 ms forward-pass latency per micro-batch. Similarly, Qwen2.5-7B required 18.00 GiB VRAM, yielding 9.38 samples/s and a 162.17 ms micro-batch latency.

6 Architectural Analysis & Ablations

To investigate the effect of Procrustes alignment on optimization behavior, we analyze the internal dynamics of the PRQ parameterization across GLUE and vision tasks, focusing on conditioning, subspace evolution, and effective-rank utilization.

6.1 Alignment vs. Unaligned Optimization

Procrustes reorientation exerts highly domain-asymmetric geometric effects. In the evaluated ViT architecture, alignment reduces the observed routing-core condition number while substantially

Task	Llama-3-8B	Qwen2.5-7B
BoolQ	80.95	69.72
PIQA	79.22	78.02
HellaSwag	59.97	59.22
WinoGrande	74.11	71.59
ARC-Easy	79.80	75.25
ARC-Challenge	51.02	45.05
OpenBookQA	35.20	32.00
7-Task Avg.	**65.75**	**61.55**

Table 6: Evaluation performance of PLORA on seven standard commonsense reasoning benchmarks, evaluated with 0-shot prompting (no in-context examples) using log-likelihood multiple-choice scoring. Both models are evaluated in 4-bit NF4 precision, using $r = 8$, an extended-rank multiplier of 16, $\alpha = 16$, and a single random seed for three epochs of Commonsense170K fine-tuning. Only PLORA parameters P, Q, and R are trainable; the pretrained backbone and language-model head remain strictly frozen.

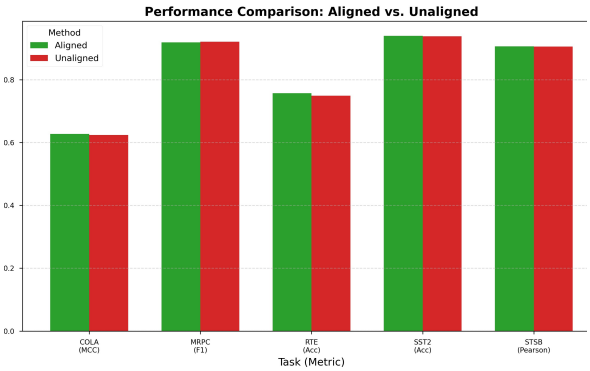


Figure 3: Performance comparison between Aligned and Unaligned configurations at $r = 16, m = 16$ on representative semantic tasks. Values are final-epoch task performance, reported as mean ± standard deviation across three random seeds.

increasing rotation of the top-r right-singular subspace, reducing minimum cosine overlap from 0.856 to 0.127 (corresponding to an increase in the maximum principal angle from approximately 31.1° to 82.7°).

In NLP, the geometric effects are task-dependent. Alignment increases the measured top-r right-subspace overlap for all four evaluated tasks, indicating less rotation of this particular subspace metric relative to the Unaligned configurations, while the condition-number response varies by task.

RTE and STS-B show relatively low adaptation strength and concentrated effective-rank utilization, with higher top-r right-subspace overlap under alignment. Reorientation therefore changes subspace evolution in a task- and metric-dependent manner: NLP conditioning remains task-dependent, while the evaluated ViT configuration shows a pronounced reduction in the conditioning proxy.

6.2 Random-Bases and No-PQ Ablations

To isolate the role of pretrained singular-vector orientations, we replace the frozen SVD-derived structural directions with random orthonormal directions (Random Bases), while retaining the fixed singular-value-derived scaling D. We also evaluate a No-PQ variant that retains only the compact core R.

Random Bases can match Aligned performance on MRPC and STS-B (Table 8), while exhibiting poorer routing-core conditioning, lower top-r right-subspace overlap, and higher effective rank (Table 9). These results highlight the role of pretrained spectral geometry in the observed optimization behavior.

The No-PQ ablation produces substantial performance degradation, dominated by a 24.67-point drop on RTE; CoLA also declines, though more modestly, by 4.19 points. This indicates that the extended routing matrices P and Q provide important adaptation capacity beyond the compact core alone. In contrast, R-Tail remains competitive with the Aligned configuration: performance decreases modestly on RTE and CoLA, while remaining comparable or slightly higher on STS-B and MRPC. Thus, the experiments suggest that extended routing capacity is more critical than strict contiguity of the selected non-principal spectral directions in the evaluated tasks. (Appendix C).

7 Discussion and Conclusion

We introduced Extended-SVD, a parameter-efficient framework separating frozen pretrained geometry from compact trainable routing dynamics. Its stabilized variant, PLORA, treats lower-energy singular directions as geometric continuations of the dominant subspace, enabling structured adaptation under aggressive parameter compression.

Across the evaluated settings, frozen spectral geometry provides a structured coordinate system for adaptation, while Procrustes-inspired reorientation changes the optimization geometry in an architecture- and task-dependent manner. In ViT, reorientation is associated with substantially lower core conditioning and greater top-r right-subspace

Task	Config.	Task Metric	$\kappa_\epsilon(R) \pm Std$	Top-r V Min. Cosine Overlap	Core Eff. Rank (R)	Adapt. Str.	Abs. Wt. Mag. ($\|W_t - W_0\|_F$)
Vision (ViT)	Aligned	92.37% ± 0.20	97.2 ± 39.2	0.127 ± 0.012	13.50 ± 0.06	0.023 ± 0.000	11.54 ± 0.06
	Unaligned	92.37% ± 0.09	1572.6 ± 1546.5	0.856 ± 0.013	10.55 ± 0.09	0.014 ± 0.000	7.30 ± 0.03
SST-2	Aligned	94.50% ± 0.53	3492 ± 2426	0.150 ± 0.014	7.29 ± 0.07	0.011 ± 0.001	9.45 ± 0.81
	Unaligned	94.00% ± 0.35	2315 ± 1650	0.108 ± 0.009	7.47 ± 0.04	0.011 ± 0.001	9.32 ± 0.64
CoLA	Aligned	63.56% ± 0.75	2635 ± 1072	0.238 ± 0.028	6.03 ± 0.11	0.009 ± 0.001	6.96 ± 0.53
	Unaligned	62.69% ± 1.93	1855 ± 411	0.182 ± 0.009	6.27 ± 0.15	0.009 ± 0.001	7.00 ± 0.57
RTE	Aligned	77.98% ± 1.08	2496 ± 763	0.444 ± 0.039	6.12 ± 0.21	0.006 ± 0.000	4.63 ± 0.26
	Unaligned	77.86% ± 0.75	2182 ± 1138	0.389 ± 0.055	6.42 ± 0.07	0.005 ± 0.001	4.31 ± 0.49
STS-B	Aligned	90.70% ± 0.16	3505 ± 906	0.360 ± 0.025	4.37 ± 0.06	0.006 ± 0.000	5.06 ± 0.30
	Unaligned	90.57% ± 0.15	4570 ± 1202	0.344 ± 0.036	4.66 ± 0.19	0.005 ± 0.001	4.54 ± 0.40

Table 7: Geometric behaviors for $r = 16$, $m = 16$ across domains and alignment configurations. Values are final-epoch statistics averaged across analyzed target layers and reported as mean ± standard deviation across three random seeds, where $\kappa_\epsilon(R)$ denotes the ϵ-regularized condition-number proxy.

Task	Metric	Random Bases	Unaligned	Aligned	Δ
RTE	Acc.	74.73±2.25	77.86±0.75	77.98±1.08	+3.25 pp
CoLA	Matt.	63.26±0.29	62.69±1.93	63.56±0.75	+0.30 pp
STS-B	Pear.	90.70±0.14	90.57±0.15	90.70±0.16	0.00 pp
MRPC	F1	92.27±0.39	92.24±0.24	91.94±0.34	−0.33 pp

Table 8: Downstream performance comparison between the Random Bases ablation variant, Unaligned Extended-SVD, and PLORA Aligned at $r = 16$, $m = 16$, with $\alpha = 16$. Random Bases uses randomly initialized orthonormal structural bases while retaining the original singular-value-derived scaling. Δ denotes the signed performance difference, in percentage points, computed as Aligned minus Random Bases. Results are averaged across three random seeds.

Task	$\kappa_\epsilon(R)$			Top-r V Overlap			Eff. Rank (R)		
	Rand.	Unalign.	Align.	Rand.	Unalign.	Align.	Rand.	Unalign.	Align.
CoLA	10,119	1,855	2,635	0.0054	0.182	0.238	6.87	6.27	6.03
RTE	3,176	2,182	2,496	0.0055	0.389	0.444	7.01	6.42	6.12
STS-B	5,455	4,570	3,505	0.0055	0.344	0.360	5.13	4.66	4.37

Table 9: Geometric comparison between Random Bases, Unaligned Extended-SVD, and Aligned PLORA at $r = 16, m = 16$. Both pretrained-SVD configurations (Unaligned and Aligned) exhibit substantially lower routing-core conditioning $\kappa_\epsilon(R)$, markedly higher top-r right-subspace overlap, and lower routing-core effective rank than Random Bases, while the conditioning shift between Unaligned and Aligned is task-dependent.

rotation, whereas NLP exhibits task-dependent conditioning and subspace behavior. The Random-Bases and No-PQ ablations further indicate that pretrained spectral structure and extended routing capacity both affect the observed optimization dynamics.

Limitations

The SVD initialization introduces a one-time upfront computational cost. While we demonstrated this cost is highly manageable at the 7–8B scale

Task	Ext-SVD ($r = 8$) (∼0.15M)	PLORA ($r = 8$) (∼0.15M)	PiSSA (∼1.32M)	MiLoRA (∼1.32M)	VeRA (∼0.04M)
SST-2	94.72	94.60	94.50	**94.95**	91.63
CoLA	**65.89**	65.10	64.53	63.86	49.68
MRPC	**93.31**	92.50	93.24	92.79	89.45
RTE	76.17	76.50	77.26	**79.06**	53.43
STS-B	**91.05**	90.90	90.95	90.89	88.04

Table 10: Performance comparison against recent SVD and projection-based PEFT methods on the GLUE benchmark (RoBERTa-base). To ensure a strictly controlled evaluation, all baseline methods were reproduced internally using the identical training framework and hyperparameter schedules as PLORA. Results report the peak performance achieved across three random seeds. Both Extended-SVD (Unaligned) and PLORA (Aligned) achieve highly competitive performance while preserving frozen pretrained geometry and utilizing 9× fewer parameters than PiSSA and MiLoRA.

Task	Metric	No-PQ	R-Tail	Unaligned	Aligned
RTE	Acc.	53.31±0.55	77.02±2.18	77.86±0.75	77.98±1.08
CoLA	Matt.	59.37±0.51	61.33±0.39	62.69±1.93	63.56±0.75
STS-B	Pear.	90.06±0.42	90.99±0.29	90.57±0.15	90.70±0.16
MRPC	F1	90.73±0.64	92.65±0.46	92.24±0.24	91.94±0.34

Table 11: Structural ablation results compared against Unaligned Extended-SVD and Aligned PLORA at $r = 16$, $m = 16$, and $\alpha = 16$. No-PQ removes the extended routing matrices P and Q, retaining only the compact core R. R-Tail replaces the contiguous mid-spectrum extension with randomly selected non-principal singular vectors. Results are averaged across three random seeds.

(e.g., ∼12 minutes on an NVIDIA A100), it may become non-negligible for extremely large weight matrices in massively parameterized frontier models (e.g., 70B+) or Mixture-of-Experts (MoE) architectures, which remain for future evaluation. Furthermore, the benefits of Procrustes reorientation are task- and architecture-dependent: the clearest conditioning reduction is observed in the evaluated ViT configuration, whereas NLP results are

task-dependent. The added structural complexity of the multi-matrix routing cascade is inherently less transparent than classical LoRA and may complicate deployment in highly optimized production systems where standard unconstrained LoRA suffices. Finally, the method's reliance on frozen SVD bases may limit adaptability in continual learning scenarios where the underlying optimal geometric subspace shifts significantly over time.

Ethical Considerations

We see no immediate ethical concerns arising from this work. The method reduces computational cost for fine-tuning large models, which may improve accessibility of NLP and vision systems. We encourage responsible use of parameter-efficient adaptation to avoid reinforcing biases present in pretrained models.

Acknowledgments

This work was carried out as part of the author's M.Tech. project at the Indian Institute of Technology Patna. The author thanks Prof. Rajiv Misra for his guidance and supervision during the project.

References

Alan Ansell, Edoardo Maria Ponti, Anna Korhonen, and Ivan Vulić. 2022. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796.

Jiaao Chen, Aston Zhang, Xingjian Shi, Mu Li, Alex Smola, and Diyi Yang. 2023. Parameter-efficient fine-tuning design spaces. In *The Eleventh International Conference on Learning Representations*.

Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, and Nando de Freitas. 2013. Predicting parameters in deep learning. In *Advances in Neural Information Processing Systems 26*, pages 2148–2156.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems 36*, pages 10088–10115.

Chuntao Ding, Xu Cao, Jianhang Xie, Linlin Fan, Shangguang Wang, and Zhichao Lu. 2024. LoRA-C: Parameter-efficient fine-tuning of robust CNN for IoT devices. *arXiv preprint arXiv:2410.16954*.

Haonan Dong, Wenhao Zhu, Guojie Song, and Liang Wang. 2025. AuroRA: Breaking low-rank bottleneck of LoRA with nonlinear mapping. In *Advances in Neural Information Processing Systems 38*, pages 41501–41533.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16×16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR 2021)*.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. 2024. Mixture-of-LoRAs: An efficient multitask tuning method for large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11371–11380.

Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. 2024. Parameter-efficient fine-tuning with discrete Fourier transform. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 14884–14901. PMLR.

Demi Guo, Alexander Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pages 770–778.

Neil Houlsby, Andrei Giurgiu, Stanisław Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR 2022)*.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria,

and Roy Lee. 2023. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5254–5276.

Yiping Ji, Hemanth Saratchandran, Cameron Gordon, Zeyu Zhang, and Simon Lucey. 2025. Efficient learning with sine-activated low-rank matrices. In *The Thirteenth International Conference on Learning Representations*.

Dawid Kopiczko, Tijmen Blankevoort, and Yuki M Asano. 2024. VeRA: Vector-based random matrix adaptation. In *International Conference on Learning Representations (ICLR 2024)*.

Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Neal Lawton, Anoop Kumar, Govind Thattai, Aram Galstyan, and Greg Ver Steeg. 2023. Neural architecture search for parameter-efficient fine-tuning of large pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8506–8515.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. 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. Association for Computational Linguistics.

Pengxiang Li, Lu Yin, Xiaowei Gao, and Shiwei Liu. 2025. Outlier-weighed layerwise sampling for LLM fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19460–19473.

Xiang Lisa Li and Percy Liang. 2021. 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. Association for Computational Linguistics.

Yinqiao Li, Linqi Song, and Hanxu Hou. 2024. LoRAN: Improved low-rank adaptation by a non-linear transformation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3134–3143.

Jing Liu, Toshiaki Koike-Akino, Pu Wang, Matthew Brand, Ye Wang, and Kieran Parsons. 2023. LoDA: Low-dimensional adaptation of large language models. In *NeurIPS'23 Workshop on Efficient Natural Language and Speech Processing*.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2024. When MOE meets LLMs: Parameter efficient fine-tuning for multi-task medical applications. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1104–1114.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Qijun Luo, Hengxu Yu, and Xiao Li. 2024. BAdam: A memory efficient full parameter optimization method for large language models. In *Advances in Neural Information Processing Systems 37*, pages 24926–24958.

Yulong Mao, Kaiyu Huang, Changhao Guan, Ganglin Bao, Fengran Mo, and Jinan Xu. 2024. DoRA: Enhancing parameter-efficient fine-tuning with dynamic rank distribution. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11662–11675.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Scott Yih, and Madian Khabsa. 2022. UniPELT: A unified framework for parameter-efficient language model tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6253–6264.

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

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

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

Adithya Renduchintala, Tugrul Konuk, and Oleksii Kuchaiev. 2024. 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. Association for Computational Linguistics.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. AdapterDrop: On the efficiency of adapters in transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946.

Steffen Schotthöfer, Emanuele Zangrando, Gianluca Ceruti, Francesco Tudisco, and Jonas Kusch. 2025. GeoLoRA: Geometric integration for parameter efficient fine-tuning. In *International Conference on Learning Representations (ICLR 2025)*, pages 23656–23685.

Yi-Lin Sung, Varun Nair, and Colin Raffel. 2021. Training neural networks with fixed sparse masks. In *Advances in Neural Information Processing Systems 34*.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2023. 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. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR 2019)*.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. 2025. MiLoRA: Harnessing minor singular components for parameter-efficient LLM finetuning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4823–4836, Albuquerque, New Mexico. Association for Computational Linguistics.

Xun Wu, Shaohan Huang, and Furu Wei. 2024. Mixture of LoRA experts. In *The Twelfth International Conference on Learning Representations*.

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

Yu Zhang, Ekapol Chuangsuwanich, and James Glass. 2014. Extracting deep neural network bottleneck features using low-rank matrix factorization. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 185–189.

Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. 2025. Neural prompt search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5268–5280.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. 2024. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 61121–61143. PMLR.

Yibo Zhong, Haoxiang Jiang, Lincan Li, Ryumei Nakada, Tianci Liu, Linjun Zhang, Huaxiu Yao, and Haoyu Wang. 2026. PEANuT: Parameter-efficient adaptation with weight-aware neural tweakers. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pages 2054–2065.

Han Zhou, Xingchen Wan, Ivan Vulić, and Anna Korhonen. 2024. AutoPEFT: Automatic configuration search for parameter-efficient fine-tuning. *Transactions of the Association for Computational Linguistics*, 12:525–542.

Hongyun Zhou, Xiangyu Lu, Wang Xu, Conghui Zhu, Tiejun Zhao, and Muyun Yang. 2025. LoRA-drop: Efficient LoRA parameter pruning based on output evaluation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5530–5543.

A Detailed Experimental Setup

A.1 Vision Training

We evaluate PLORA on ResNet-50 and ViT-Base/16 using CIFAR-100. All models are trained for 30 epochs with a batch size of 16. For ResNet-50, we use a learning rate of 3×10^{-3}, weight decay of 0.02, and a 2-epoch linear warmup. For ViT-Base/16, we use a learning rate of 5×10^{-4}, weight decay of 0.01, and a 10-epoch warmup. Both architectures employ cosine learning rate decay and gradient clipping with max_norm $= 1.0$.

A.2 NLP Training

We fine-tune RoBERTa-base across the GLUE benchmark. Optimization is performed using AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. To prevent adapter degradation, we strictly enforce a weight decay of 0.0 for all PLORA modules. All tasks utilize a linear warmup ratio of 0.06 (6% of total steps).

Due to dataset variances, we apply size-dependent hyperparameter scheduling coupled with early stopping (patience of 5 to 10 epochs):

- **Small/Medium Datasets** (CoLA, SST-2, MRPC, RTE, STS-B): Learning rate of 2×10^{-4}. Maximum epochs are set to 35 for CoLA and STS-B, 30 for MRPC and RTE, and 15 for SST-2. Batch sizes vary between 16 and 32 (with gradient accumulation used to normalize the effective batch size).
- **Large Datasets** (MNLI, QNLI): Learning rate of 1×10^{-4}. Maximum epochs are set to 15, with a batch size of 64.

A.3 Experimental Setup for LLMs

Gemma-2B Setup (GLUE Evaluation):

- **Base Model:** Gemma-2B (loaded in 4-bit NF4 quantization).
- **PLORA Configuration:** Base rank $(r) = 16$, Extended rank multiplier $(m) = 16$, Alpha $(\alpha) = 16.0$.
- **Training Hyperparameters:** Learning rate $= 2 \times 10^{-4}$, Effective batch size $= 16$ (physical batch size 4 with 4 gradient accumulation steps), Maximum sequence length $= 128$.
- **Optimization:** Models were trained for up to 10 epochs using the Paged AdamW 8-bit optimizer with early stopping (patience $= 3$). The best model checkpoint was selected based on the primary validation metric for each respective task (RTE, MRPC, CoLA, STS-B).

Llama-3-8B and Qwen2.5-7B Setup (Commonsense170K Evaluation):

- **PLORA Configuration:** Base rank $(r) = 8$, Extended rank multiplier $(m) = 16$, Alpha $(\alpha) = 16.0$, with a dropout of 0.05. The adaptation targeted all dense linear projections (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`).
- **Optimization:** We utilized the Paged AdamW 32-bit optimizer with a learning rate of 3×10^{-4}, a weight decay of 0.0, and 120 linear warmup steps.
- **Batching & Context:** Models were trained with a maximum sequence length of 256, using a physical per-device batch size of 16 and 8 gradient accumulation steps (effective batch size 128).
- **Hardware Telemetry:** Training was performed on a single NVIDIA A100 80GB GPU. Telemetry (measured using a 128-token context sequence) confirms the physical execution profile: Llama-3-8B required 17.54 GiB peak allocated VRAM, achieving 9.01 samples/s throughput and 215.76 ms forward-pass latency per micro-batch. Similarly, Qwen2.5-7B required 18.00 GiB VRAM, yielding 9.38 samples/s and a 162.17 ms micro-batch latency.
- **Evaluation Protocol (`lm-eval`):** The SVD initialization was performed dynamically on the unquantized float16 weights prior to 4-bit NF4 quantization. Evaluations were conducted using the EleutherAI LM Evaluation Harness (v0.4.2) with `num_fewshot=0`. Exact task identifiers used were `boolq`, `piqa`, `hellaswag`, `winogrande`, `arc_easy`, `arc_challenge`, and `openbookqa`, evaluated on their standard default splits (e.g., `validation` or `test` as defined by the harness).

B Architectural Metric Definitions (Code-Consistent Formulation)

To analyze the optimization dynamics of the Extended-SVD framework, we track several geometric and numerical metrics throughout training. All metrics are computed directly from the learned routing matrices and the resulting composite update matrix. Reported geometric metrics are computed per target layer and averaged across the analyzed layers; task-level values are then aggregated across random seeds.

B.1 Effective Rank

For a matrix M with singular values $\{\sigma_i\}$, we define the normalized singular-value distribution

$$p_i = \frac{\sigma_i}{\sum_j \sigma_j}. \tag{3}$$

For the zero matrix, effective rank is defined as 0; subsequent values use the entropy-equivalent rank of the normalized singular-value distribution:

$$\text{rank}_{\text{eff}}(M) = \exp\left(-\sum_i p_i \log(p_i + \varepsilon)\right), \tag{4}$$

where ε is a small numerical constant for stability.

Notation	Definition
$\mathcal{W}_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$	Frozen pre-trained weight matrix.
$\Delta W_{\text{raw}} \in \mathbb{R}^{d_{out} \times d_{in}}$	Unscaled composite weight update matrix.
$\Delta W_{\text{app}} \in \mathbb{R}^{d_{out} \times d_{in}}$	Applied (scaled) composite weight update matrix.
r	Core adaptation rank controlling the dimensionality of the compact mixing component.
m	Extension multiplier (determines the size of the extended bases).
r_{ext}	Extended subspace dimension, defined as $r_{\text{ext}} = r \cdot m$.
$D \in \mathbb{R}^{r_{\text{ext}} \times r_{\text{ext}}}$	Fixed diagonal scaling matrix containing the square roots of extended singular values.
$U_c \in \mathbb{R}^{d_{\text{out}} \times r_{\text{ext}}}$	Frozen left structural matrix constructed from the SVD-derived blocks and the Procrustes-inspired transformation.
$V_c \in \mathbb{R}^{d_{\text{in}} \times r_{\text{ext}}}$	Frozen right structural matrix constructed from the SVD-derived blocks and the Procrustes-inspired transformation.
$R \in \mathbb{R}^{r \times r}$	Learnable core mixing matrix (dense feature adaptation).
$P \in \mathbb{R}^{r_{\text{ext}} \times r}$	Learnable projection matrix mapping from the compact core into the extended adaptation subspace.
$Q \in \mathbb{R}^{r \times r_{\text{ext}}}$	Learnable projection matrix mapping from the extended adaptation subspace back into the compact core.
α	Constant scaling factor.
$\kappa_\epsilon(R)$	ϵ-regularized condition-number proxy of the core mixing matrix (numerical stability metric).
$x \in \mathbb{R}^{d_{\text{in}}}, h \in \mathbb{R}^{d_{\text{out}}}$	Input vector and output hidden state vector, respectively.

Table 12: Summary of mathematical notations used in the Extended-SVD architecture.

Unlike the algebraic rank, the effective rank measures how broadly the singular-value energy is distributed across dimensions. Lower values indicate more concentrated adaptation dynamics.

B.2 Condition Number

To handle initialization states where $R = 0$, we define an ϵ-regularized condition-number proxy:

$$\kappa_\epsilon(M) = \frac{\sigma_{max}(M)}{\sigma_{min}(M) + \epsilon} \tag{5}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ are the largest and smallest singular values, respectively.

Within this framework, we primarily track the conditioning of the compact routing core R:

$$\kappa_\epsilon(R). \tag{6}$$

Large values indicate greater numerical anisotropy and ill-conditioning of the routing core R.

B.3 Weight Change and Adaptation Strength

To quantify the magnitude of the learned updates, we define two distinct metrics based on the unscaled adapter update ΔW_{raw}:

Adaptation Strength measures the relative norm of the unscaled update:

$$\text{Adaptation Strength} = \frac{\|\Delta W_{\text{raw}}\|_F}{\|W_0\|_F} \tag{7}$$

Applied Relative Weight Change measures the relative change actually applied to the pretrained weight. Because the full adapted weight is $W_t = W_0 + \Delta W_{\text{app}}$, this metric scales strictly by the effective multiplier λ used in the respective implementation:

$$\begin{aligned} \text{Applied Rel. Wt. Change} &= \frac{\|\Delta W_{\text{app}}\|_F}{\|W_0\|_F} \\ &= \lambda \frac{\|\Delta W_{\text{raw}}\|_F}{\|W_0\|_F} \end{aligned} \tag{8}$$

For the primary Aligned and Unaligned configurations and the structural ablations Random-Bases, Random-Tail, and No-PQ, $\lambda = \alpha$. PiSSA and MiLoRA use $\lambda = 1$ for their native AB updates, while VeRA retains its native module scaling. For the primary $r = 16, \alpha = 16$ experiments, $\lambda = \alpha = 16$, meaning the applied change is exactly $16\times$ the raw Adaptation Strength.

B.4 Extended Routing Emphasis

To measure how strongly optimization utilizes the extended routing dimensions beyond the principal routing region associated with the dominant singular directions, we define the Extended Routing Emphasis ratios for P and Q.

Let $r_{\text{ext}} = r \cdot m$. The projection matrix $P \in \mathbb{R}^{r_{\text{ext}} \times r}$ is partitioned into:

- the principal block: $P_{\text{core}} = P_{1:r,:}$,
- and the extended block: $P_{\text{ext}} = P_{r+1:r_{\text{ext}},:}$.

The Extended Routing Emphasis for P is then

$$E_P = \frac{\|P_{\text{ext}}\|_F}{\|P_{\text{core}}\|_F + \epsilon}. \tag{9}$$

Similarly, for $Q \in \mathbb{R}^{r \times r_{\text{ext}}}$, we define $Q_{\text{core}} = Q_{:,1:r}$, $Q_{\text{ext}} = Q_{:,r+1:r_{\text{ext}}}$, and compute

$$E_Q = \frac{\|Q_{\text{ext}}\|_F}{\|Q_{\text{core}}\|_F + \epsilon}. \tag{10}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and ϵ is a small constant added for numerical stability.

These metrics quantify the relative contribution of the extended routing dimensions compared to the compact routing region. Higher values indicate greater relative Frobenius magnitude in the extended routing block.

B.5 PRQ Effective Rank

To characterize the effective dimensionality of the internal routing cascade, we compute the effective rank of the composite routing matrix PRQ. Using the same entropy-based definition,

$$\text{rank}_{\text{eff}}(PRQ) = \exp\left(-\sum_i p_i \log(p_i + \varepsilon)\right), \tag{11}$$

where p_i are normalized singular values of PRQ.

This metric measures how broadly the compact routing dynamics utilize the available low-rank adaptation capacity.

B.6 Top-r V Minimum Cosine Overlap

Let $W_0 = U_0 \Sigma_0 V_0^\top$ and $W_t = U_t \Sigma_t V_t^\top$. Let $V_{0,r}$ and $V_{t,r}$ denote the top-r right-singular subspaces of the pretrained and adapted weights. To track local routing dynamics, we define the minimum cosine overlap as:

$$C_V(t) = \sigma_{\min}(V_{0,r}^T V_{t,r}) \tag{12}$$

The corresponding maximum principal angle is $\theta_{V,\max}(t) = \arccos C_V(t)$. Higher C_V indicates greater preservation of the pretrained right-singular subspace, whereas lower C_V indicates greater subspace rotation.

B.7 Full-Weight Top-k Left-Singular-Subspace Mean Principal Angle

We measure full-weight spectral-subspace changes between the pretrained weight W_0 and the adapted weight $W_t = W_0 + \Delta W_{\text{app}}$. After reshaping each weight tensor to a matrix, we compute the top-k left singular subspaces, with $k = \min(50, \min(C_{out}, C_{in}))$, and report the mean principal angle between these subspaces in radians.

B.8 R Diagonal Dominance

To analyze the structure of the compact routing core R, we measure its diagonal dominance:

$$D_R = \frac{\|\text{diag}(R)\|_2}{\|R - \text{diag}(\text{diag}(R))\|_F + \varepsilon}. \tag{13}$$

Higher values indicate that the routing dynamics remain closer to axis-aligned transformations, whereas lower values indicate stronger cross-dimensional mixing inside the compact adaptation core.

B.9 Tabular Glossary Metrics

In the results tables, Task Metric refers to the final downstream task performance metric accompanied by its standard deviation across random seeds. Top-r V Minimum Cosine Overlap denotes $C_V = \sigma_{\min}(V_{0,r}^T V_{t,r})$, where lower values indicate greater rotation of the pretrained right-singular subspace. Core Eff. Rank denotes the entropy-based effective rank of the compact routing matrix R. Adapt. Str. denotes the relative Frobenius norm of the unscaled update ΔW_{raw} with respect to the pretrained weight. Abs. Wt. Mag. denotes $\|W_t - W_0\|_F$. Finally, in all condition-number computations, $\epsilon = 10^{-10}$ is used for numerical stability during the initial $R \leftarrow 0$ state.

C Geometric Conditioning Under Procrustes-Inspired Reorientation

Procrustes-inspired coordinate reorientation influences optimization by modifying the coordinate orientation within the frozen extended subspace. Empirically, its effect differs across domains. In vision architectures under aggressive compression, alignment frequently improves conditioning and stabilizes optimization. In NLP, alignment changes optimization trajectories, but its effects on conditioning, seed variability, and peak task performance are task-dependent.

Because U_c and V_c are frozen structural bases and D is fixed, the conditioning behavior is governed primarily by the trainable routing cascade P, R, Q. Unaligned configurations often require larger corrective transformations in the routing projections, which can produce substantially larger condition numbers during training. Figure 2 shows a substantially lower condition-number proxy for the Aligned configuration in the evaluated ViT setting. Table 7 indicates task-dependent behavior in NLP: the Aligned proxy is lower for STS-B

but higher for SST-2, CoLA, and RTE. Thus, the conditioning results do not support a uniform improvement across tasks.

A modest mechanistic interpretation is that coordinate reorientation changes the association of the near-boundary extension vectors with the fixed spectral scaling at initialization. This may reduce the geometric correction required from the trainable routing projections. This does not change the rank limit of the update, but it can change how learning is distributed across P, R, and Q. The observed effects on variance, subspace drift, and effective-rank utilization are consistent with this view.

Additional telemetry reveals that the compact routing core remains the primary adaptation bottleneck even when the extended routing space grows substantially. While the effective rank of the internal routing matrix R increases with adaptation rank, the effective rank of the composite cascade PRQ grows much more slowly and remains highly compressed relative to the nominal update dimension. This suggests that the extended basis primarily acts as a structured routing subspace, whereas the compact core controls most of the effective mixing dynamics. Empirically, this behavior supports the interpretation that coordinate reorientation modifies how optimization trajectories utilize the extended adaptation subspace. At $r = 16$ and $m = 16$, the final-epoch Extended Routing Emphasis ratios show that the extended projection blocks retain substantial relative Frobenius magnitude compared with their principal blocks. These endpoint measurements characterize the learned parameterization at convergence but do not, by themselves, establish how much P or Q changed during optimization. Taken together with the effective-rank measurements, the results are consistent with P and Q providing extended routing coordinates while R remains the compact rank bottleneck.

C.1 Geometric Interpretation of Extended-SVD Adaptation

The Extended-SVD architecture constrains the learned routing cascade to a fixed pretrained spectral coordinate system. The applied update matrix is parameterized as

$$\Delta W_{\text{app}} = \alpha U_c D P R Q D V_c^T \tag{14}$$

where U_c and V_c are the frozen SVD-derived structural matrices obtained after the Procrustes-inspired block transformation, D is the fixed diagonal singular-value-derived scaling matrix, and $U_{\text{base}} = U_c D$ and $V_{\text{base}} = V_c D$ are the corresponding frozen structural factors. Only the routing matrices P, R, and Q are trained; the structural matrices and scaling remain fixed throughout optimization.

Unlike standard LoRA, where the low-rank factors directly determine both the geometry and magnitude of the update, the Extended-SVD framework separates:

- the adaptation coordinate system and singular-value-derived scaling (frozen structural factors),
- from the task-specific transformation dynamics (routing matrices).

This separation constrains optimization to occur within a fixed pretrained spectral neighborhood.

C.2 Proposition 1: Rank Preservation

Because $R \in \mathbb{R}^{r \times r}$, the applied composite update satisfies $\text{rank}(\Delta W_{\text{app}}) \leq r$.

Proof. Using the submultiplicativity of matrix rank, $\text{rank}(ABC) \leq \min(\text{rank}(A), \text{rank}(B), \text{rank}(C))$, we obtain $\text{rank}(U_{\text{base}} P R Q V_{\text{base}}^T) \leq \text{rank}(R) \leq r$. Thus, despite operating inside an extended subspace of dimension r_{ext}, the effective update rank remains bounded by the compact routing core.

C.3 Proposition 2: Extended-SVD Defines a Constrained Adaptation Family

Define the Extended-SVD adaptation family $\mathcal{M} = \{U_{base} P R Q V_{base}^T : P, Q, R\}$. Then all structural updates lie inside the fixed subspace $\text{span}(U_c) \times \text{span}(V_c)$.

Interpretation. The framework therefore restricts optimization to a pretrained spectral coordinate system rather than allowing unrestricted low-rank rotations during training. This constraint does not reduce the nominal rank capacity relative to standard rank r adaptation, but it fundamentally limits the admissible update geometry. As established in our empirical analysis, this geometric constraint manifests as a domain-asymmetric regularization effect:

- highly concentrated update spectra (lower effective rank) in language tasks,

- reduced observed routing-core condition-number growth and larger measured full-weight modification in the evaluated vision configuration,
- and modality-specific optimization trajectories, with the measured top-r right-subspace response remaining task- and metric-dependent.

C.4 Empirical Observation on Conditioning (Geometry-Induced Conditioning Control)

Because U_c, V_c, and D remain frozen during optimization, the trainable routing cascade P, R, and Q evolves within a fixed SVD-derived parameterization. The implemented Procrustes-inspired transformation does not guarantee that U_c and V_c are orthonormal; accordingly, we interpret its effects as empirical properties of the complete parameterization rather than consequences of a guaranteed orthogonal transformation.

In the evaluated NLP tasks, reorientation does not consistently reduce the routing-core condition-number proxy. However, the measured top-r right-subspace overlap is higher for the Aligned configuration across the evaluated tasks, indicating greater preservation of that particular subspace according to the defined metric. The aligned configurations also exhibit lower routing-core effective rank in the reported comparisons. These observations support a task-dependent, geometry-aware interpretation of reorientation, rather than a theoretical guarantee of bounded or uniformly improved conditioning.

C.5 Architectural Conclusion

Random Bases can retain competitive task performance despite substantially different internal optimization geometry. However, the results in Table 9 show that randomizing the pretrained singular-vector directions while retaining the fixed scaling D consistently leads to less favorable routing geometry than the Aligned SVD configuration:

1. **Routing-Core Conditioning Under Random Bases:** Because Random Bases replace the pretrained singular-vector directions with random orthonormal directions while retaining the fixed scaling D, the trainable routing matrices (P, R, Q) operate in a coordinate system whose directions are less aligned with the pretrained weight structure. Empirically, Random Bases produce larger observed routing-core condition-number proxies $\kappa_\epsilon(R)$ than the Aligned configuration across the reported NLP tasks. The magnitude of this degradation is task-dependent.
2. **Top-r Subspace Preservation:** Random Bases exhibit substantially lower top-r right-subspace overlap than Aligned PLORA. The minimum cosine overlap is approximately 0.0055 for the Random-Bases configurations, compared with 0.238–0.444 for the corresponding Aligned configurations. This indicates that the adapted weight's top-r right-singular subspace is much less aligned with the pretrained principal right-singular subspace, corresponding to substantially greater subspace rotation.
3. **Routing-Core Effective Rank:** Random Bases also produce higher routing-core effective rank across the reported tasks, indicating less concentrated utilization of the routing coordinates. Together with the conditioning and subspace-overlap results, this suggests that pretrained spectral bases provide a more structured coordinate system for parameter-efficient adaptation.

To further quantify these internal dynamics, we track the composite effective rank of the routing cascade ($\text{rank}_{\text{eff}}(PRQ)$), the diagonal dominance of the core matrix (D_R), and full-weight drift. As shown in Table 14, while the extended adaptation subspace dimension is large ($r_{\text{ext}} = 256$), the learned routing cascade exhibits strongly compressed effective dimensionality. The diagonal dominance scores ($D_R \approx 0.24 - 0.26$) indicate that the learned core is not strongly diagonally dominated (matching the ≈ 0.258 theoretical reference scale for a dense equal-variance matrix). Finally, the full-weight Top-k Left-Singular-Subspace mean principal angle quantifies the observed change in the leading spectral subspace, with a larger shift observed in ViT (0.270 rad) than in the reported NLP tasks (0.044–0.078 rad). (The step-by-step optimization trajectories and layer-wise training dynamics corresponding to these configurations are visualized in Figures 4 through 9).

C.6 The Functional Role of Extended Routing Dimensions

A natural concern in extended low-rank adaptation is whether the additional projection matrices P

Task	Config.	Core Eff. Rank (R)	Adapt. Str.	Abs. Wt. Mag.	E_P	E_Q
SST-2	Aligned	7.29 ± 0.07	0.0113 ± 0.0009	9.45 ± 0.81	4.13 ± 0.06	4.72 ± 0.02
	Unaligned	7.47 ± 0.04	0.0111 ± 0.0008	9.32 ± 0.64	4.10 ± 0.04	4.73 ± 0.04
RTE	Aligned	6.12 ± 0.21	0.0058 ± 0.0002	4.63 ± 0.26	3.99 ± 0.04	4.09 ± 0.02
	Unaligned	6.42 ± 0.07	0.0053 ± 0.0007	4.31 ± 0.49	4.01 ± 0.02	4.06 ± 0.09
STS-B	Aligned	4.37 ± 0.06	0.0061 ± 0.0003	5.06 ± 0.30	4.02 ± 0.06	4.11 ± 0.04
	Unaligned	4.66 ± 0.19	0.0054 ± 0.0006	4.54 ± 0.40	4.01 ± 0.03	4.08 ± 0.02
CoLA	Aligned	6.03 ± 0.11	0.0086 ± 0.0006	6.96 ± 0.53	4.18 ± 0.04	4.41 ± 0.08
	Unaligned	6.27 ± 0.15	0.0087 ± 0.0007	7.00 ± 0.57	4.24 ± 0.05	4.44 ± 0.02

Table 13: Extended Routing Emphasis and core dynamics across NLP tasks ($r = 16, m = 16$). Values are final-epoch statistics averaged across analyzed target layers and reported as mean ± standard deviation across three random seeds.

Task	Config.	Composite Eff. Rank (PRQ)	Core Diag. Dominance (D_R)	Full-Weight Drift
Vision (ViT)	Aligned	9.69 ± 0.06	0.240 ± 0.024	0.270 ± 0.002
	Unaligned	7.11 ± 0.13	0.244 ± 0.013	0.147 ± 0.003
SST-2	Aligned	2.14 ± 0.06	0.255 ± 0.004	0.074 ± 0.004
	Unaligned	2.25 ± 0.09	0.253 ± 0.003	0.078 ± 0.004
CoLA	Aligned	1.73 ± 0.06	0.257 ± 0.003	0.056 ± 0.001
	Unaligned	1.83 ± 0.08	0.256 ± 0.006	0.059 ± 0.002
RTE	Aligned	1.95 ± 0.07	0.255 ± 0.003	0.050 ± 0.001
	Unaligned	2.04 ± 0.04	0.257 ± 0.005	0.050 ± 0.002
STS-B	Aligned	1.34 ± 0.02	0.256 ± 0.005	0.044 ± 0.001
	Unaligned	1.44 ± 0.03	0.255 ± 0.003	0.044 ± 0.001

Table 14: Internal routing cascade dynamics ($r = 16, m = 16$). The highly compressed composite PRQ effective rank indicates strongly compressed effective dimensionality in the learned routing cascade. The observed diagonal dominance indicates that the learned core is not strongly diagonally dominated (matching the ≈ 0.258 theoretical reference scale for a dense equal-variance matrix). Full-weight drift measures the observed structural modification of the pretrained weight geometry. Values are final-epoch statistics averaged across analyzed target layers and reported as mean ± standard deviation across three random seeds.

and Q become redundant when the core matrix R enforces a strict bottleneck. The Extended Routing Emphasis metrics quantify the relative Frobenius magnitude of the extended and principal blocks of P and Q.

Across the four NLP tasks reported in Table 13, both P and Q maintain final-epoch emphasis ratios in the range of approximately $4.0\times$ to $4.7\times$. With $r = 16$ and $m = 16$, the expected equal-variance Frobenius norm ratio between the extended block (240×16) and the principal block (16×16) is

$$\sqrt{240/16} = \sqrt{15} \approx 3.87.$$

The observed final ratios (3.99–4.73) remain close to this equal-variance reference, indicating that the extended blocks retain substantial relative Frobenius magnitude at convergence. Because Table 13 reports final-epoch ratios only, these measurements do not by themselves establish how much the extended blocks changed during optimization.

The resulting decomposition can therefore be interpreted as follows:

- The compact matrix (R) serves as the primary low-dimensional mixing component, controlling the bottlenecked transformation.
- The surrounding projection matrices (P and Q) retain substantial final-epoch relative magnitude within the extended frozen subspace, providing the parameterization through which routing is expressed.

At the final epoch, the extended routing matrices

P and Q remain close in scale to the equal-variance reference ratio. Procrustes reorientation does not substantially change these relative magnitudes; instead, it changes the parameterization of the extended projection blocks and is associated with a modest reduction in the effective rank of the core matrix (e.g., $7.47 \rightarrow 7.29$ on SST-2), while maintaining comparable Extended Routing Emphasis ratios.

Taken together, these results indicate that the learned parameterization retains substantial relative magnitude in the extended projection blocks, while the effective transformation remains strongly bottlenecked through the compact core R. The measurements therefore support a role for P and Q as extended routing coordinates, but do not by themselves establish the extent of their optimization-induced movement from initialization.

C.7 Algorithm Workflow and Static Merging

During training, the Extended-SVD framework operates as a structured five-matrix cascade, with the applied update $\Delta W_{\text{app}} = \alpha \Delta W_{\text{raw}} = \alpha U_{\text{base}} P R Q V_{\text{base}}^T$, where the frozen bases U_{base} and V_{base} define the adaptation coordinate system, while the trainable matrices $\{P, R, Q\}$ control task-specific routing dynamics. Under aggressive compression, this structured parameterization can improve optimization stability and reduce condition-number growth, particularly in vision architectures.

After training, the cascade is merged into a single static applied update matrix, $\Delta W_{\text{app}} = \alpha U_{\text{base}} P R Q V_{\text{base}}^T$, which is permanently added to the pretrained weights: $W_{\text{inf}} = W_0 + \Delta W_{\text{app}}$. Because the transformation is fully linear, inference introduces no additional computational or memory overhead relative to the base model.

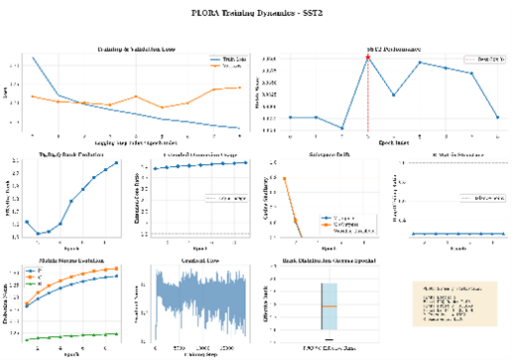


Figure 4: Aligned PLORA configurations on SST-2 under identical capacity ($r = 16, m = 16$).

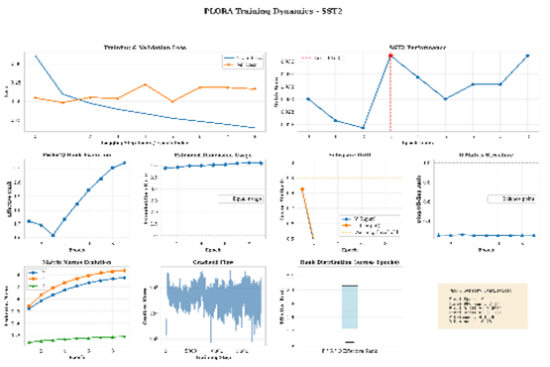


Figure 5: Unaligned Extended-SVD configurations on SST-2 under identical capacity ($r = 16, m = 16$).

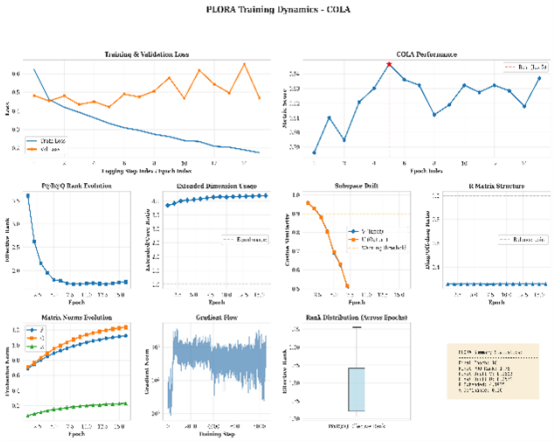


Figure 6: PLORA training dynamics on CoLA ($r = 16, m = 16$), Aligned.

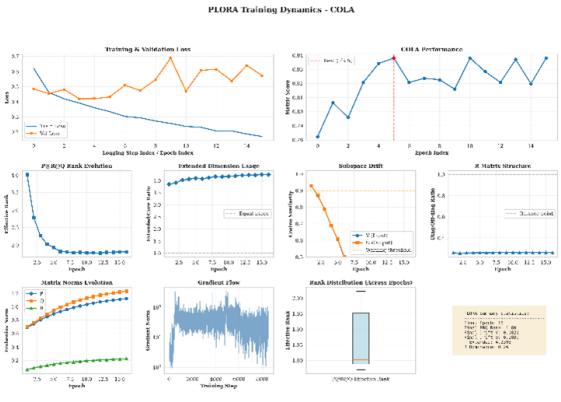


Figure 7: Extended-SVD training dynamics on CoLA ($r = 16, m = 16$), Unaligned.

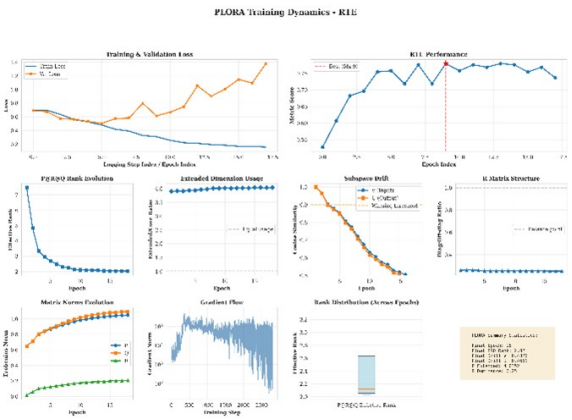


Figure 8: PLORA training dynamics on RTE ($r = 16, m = 16$), Aligned.

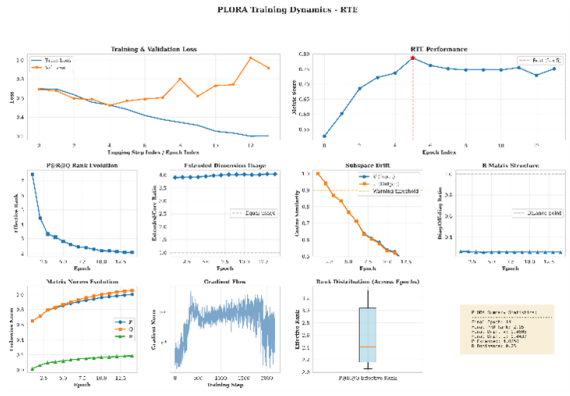


Figure 9: Extended-SVD training dynamics on RTE ($r = 16, m = 16$), Unaligned.

Algorithm 1 Extended-SVD Training and Merge Procedure (with Optional PLORA Reorientation)

Require: Pretrained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, rank r, extension multiplier m, scaling factor α, training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, epochs T.

Phase 1: Initialization

$r_{\text{ext}} \leftarrow r \cdot m$

Compute thin SVD of W_0:

$$W_0 \approx U\Sigma V^T$$

Extract the leading r_{ext} components and optionally apply the Procrustes-inspired block transformation to construct the frozen structural matrices U_c and V_c.

Construct fixed spectral scaling matrix $D \in \mathbb{R}^{r_{\text{ext}}\times r_{\text{ext}}}$ from singular values

Form frozen structural factors:

$$U_{\text{base}} \leftarrow U_c D, \quad V_{\text{base}} \leftarrow V_c D$$

Initialize trainable routing matrices:

$$P \in \mathbb{R}^{r_{\text{ext}}\times r}, \quad R \in \mathbb{R}^{r\times r}, \quad Q \in \mathbb{R}^{r\times r_{\text{ext}}}$$

Initialize compact core matrix: $R \leftarrow 0$ so that initially $\Delta W_{\text{raw}} = 0$

Phase 2: Training

for each epoch $t = 1 \dots T$ **do**

  for each minibatch $(x, y) \in \mathcal{D}$ **do**

    Compute unscaled low-rank update:

$$\Delta W_{\text{raw}} = U_{\text{base}} PRQ V_{\text{base}}^T$$

    $h = W_0 x + \alpha \Delta W_{\text{raw}} x$

    Compute loss: $\mathcal{L}(h, y)$

    Update only (P, R, Q) using backpropagation

  end for

end for

Phase 3: Static Merging

Merge trained adapter into static weight:

$$W_{\text{final}} = W_0 + \alpha\, U_{\text{base}} PRQ V_{\text{base}}^T$$

return W_{final}