HRHA: A HIGH RANK AND HIGH APPLICABILITY METHOD FOR PARAMETER-EFFICIENT FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Parameter-efficient fine-tuning (PEFT) is essential for adapting foundation models, yet the widely adopted low-rank paradigm imposes an explicit rank constraint that couples parameter efficiency with adaptation flexibility. This becomes a bottleneck when tasks, models, and layers exhibit heterogeneous intrinsic ranks and spectral structures, as no single small rank is universally sufficient. We propose **HRHA**, a **H**igh-**R**ank and **H**igh-**A**pplicability PEFT method that decouples parameter efficiency from low-rank parameterization. HRHA partitions each weight update into non-overlapping blocks and represents them with an SVD-inspired parameterization: left and right spectral bases are shared and frozen within each layer, while only block-specific diagonal coefficients are learned. Parameter efficiency is thus achieved through *structured parameter sharing* rather than rank reduction, and the resulting update is not explicitly rank-constrained, allowing full-rank or near-full-rank adaptation with substantially fewer trainable parameters than full fine-tuning. We theoretically establish that HRHA imposes no additional rank constraint, and empirically show that it attains near-full numerical rank and higher stable and effective ranks than representative high-rank LoRA variants. Extensive experiments on five benchmarks across diverse architectures and scales demonstrate consistent, stable, and parameter-efficient performance, highlighting the broad applicability of HRHA.

1 INTRODUCTION

As pre-trained models continue to scale, parameter-efficient fine-tuning (PEFT) has become essential for adapting foundation models under limited computational and memory budgets He et al. (2021). Among existing approaches, Low-Rank Adaptation (LoRA) Hu et al. (2022) has become a widely adopted paradigm. LoRA represents the weight update as the product of two low-rank matrices, substantially reducing the number of trainable parameters. Despite its remarkable efficiency, LoRA relies on a fundamental structural assumption: the desired weight update can be well approximated by a low-rank matrix. This assumption is attractive for parameter reduction, but it also introduces a fundamental limitation on the *applicability* of PEFT. When the required adaptation is intrinsically high-rank or exhibits a complex spectral structure, increasing the rank becomes necessary to improve expressiveness, yet doing so directly increases the trainable parameter budget. Thus, conventional low-rank adaptation faces an inherent trade-off between *parameter efficiency* and *adaptation flexibility*. This limitation becomes particularly important when a single PEFT method is expected to operate across diverse tasks and pre-trained models. Different downstream tasks can induce substantially different weight-update patterns: some may be adequately captured by a small number of dominant directions, whereas others may require a much broader set of directions. Similarly, different pre-trained models and layers can exhibit different intrinsic ranks, spectral decay rates, and parameter redundancies. Therefore, there may not exist a single small rank that is simultaneously sufficient across tasks, models, and layers. A rank that is effective in one setting may become overly restrictive in another, while increasing the rank to accommodate the latter undermines the parameter efficiency that motivates low-rank adaptation in the first place. From this perspective, the low-rank assumption is not merely a limitation on the numerical rank of the update; it constitutes a structural bottleneck that can make PEFT methods sensitive to the characteristics of the target task and model.

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.

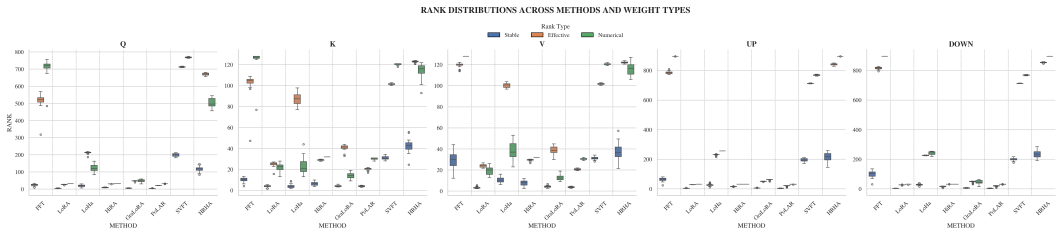

Figure 1: Rank statistics of FFT, LoRA, LoHA, GraLoRA, PoLAR, SVFT, and HRHA. Results are obtained on Qwen2.5-0.5B for the commonsense reasoning task.

Several recent methods, including LoHa Hyeon-Woo et al. (2021), HiRA Huang et al. (2025), GraLoRA Jung et al. (2026), and PoLAR Lion et al. (2026), attempt to increase the rank of the learned update while retaining parameter efficiency. However, as shown in Fig. 1, these methods can still exhibit relatively low stable and effective ranks, indicating that their spectral energy remains concentrated in a limited number of directions. SVFT Lingam et al. (2024) achieves substantially higher ranks by learning sparse coefficients over the singular vectors of the pre-trained weights, but employs a global spectral parameterization that may restrict heterogeneous adaptation across different regions of a weight matrix. These observations raise a fundamental question: *Can we achieve parameter-efficient fine-tuning without imposing an explicit low-rank constraint, while remaining applicable across diverse tasks and models?*

We answer this question with **HRHA**, a **H**igh-**R**ank and **H**igh-**A**pplicability PEFT method that decouples parameter efficiency from low-rank parameterization. Instead of representing the entire update through a global low-rank factorization, HRHA partitions the update into non-overlapping blocks and parameterizes each block using an SVD-inspired representation. The left and right spectral bases are shared and frozen within each layer, while only block-specific diagonal coefficients are learned. Thus, parameter efficiency is achieved through *structured parameter sharing* rather than rank reduction, while the block-wise coefficients provide flexibility for heterogeneous adaptation. A key property of HRHA is that it **imposes no explicit rank constraint on the resulting weight update**. The resulting update can therefore be full-rank or near-full-rank without requiring the parameter budget of full fine-tuning. This removes the need to select a small rank that simultaneously fits different tasks and models, making HRHA less dependent on their intrinsic rank and spectral characteristics. As shown in Fig. 1, HRHA achieves near-full numerical rank and substantially higher stable and effective ranks than representative high-rank LoRA variants, with a more balanced spectral distribution. More importantly, these properties are consistently observed across different downstream tasks and model architectures. We extensively evaluate HRHA on five benchmarks using models with different architectures and scales. Across these diverse settings, HRHA consistently achieves strong performance with substantially fewer trainable parameters. Moreover, its performance remains stable across tasks and model architectures, demonstrating the high applicability of HRHA.

In summary, our main contributions are: 1) We propose **HRHA**, a high-applicability PEFT method that achieves parameter efficiency through shared spectral bases rather than an explicit low-rank constraint, enabling flexible region-wise adaptation. 2) We theoretically establish that **HRHA imposes no additional rank constraint** on the fine-tuning update, and empirically demonstrate its near-full numerical rank and substantially higher stable and effective ranks. 3) Extensive experiments across diverse tasks, model architectures, and scales demonstrate the **broad applicability and parameter efficiency of HRHA**, showing that rank-unconstrained adaptation can provide strong and consistent performance across different settings.

2 METHOD

This section presents the full methodology of HRHA. We first introduce its block-wise SVD-inspired parameterization, which represents the weight update using shared spectral bases and block-specific coefficients. We then establish the rank properties of HRHA, showing that the proposed parameterization does not impose an explicit low-rank constraint and can attain the maximum possible

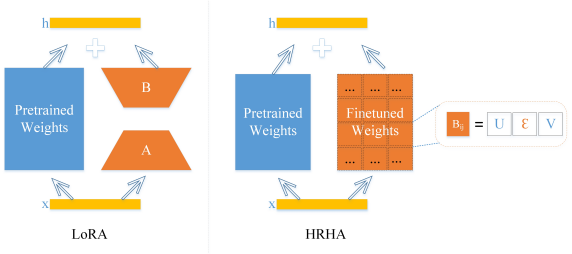

Figure 2: The frameworks of LoRA and HRHA differ fundamentally in their parameterization strategies. LoRA models fine-tuned weight updates as the product of two low-rank matrices, thereby imposing a low-rank constraint on the weight update matrix. HRHA represents fine-tuned weights as a composition of non-overlapping block matrices.

rank. On this basis, we discuss how HRHA reduces the dependence of the adaptation space on task- and model-specific rank assumptions. Finally, we analyze the optimization properties of HRHA and show that the resulting parameterization is linear in the trainable parameters, thereby yielding favorable optimization properties.

2.1 Theory of HRHA

Following LoRA, we adopt the modified forward pass $h = W_0 x + \Delta W x$, where W_0 denotes the frozen pre-trained weight matrix and ΔW is the trainable update. The core idea of HRHA is to achieve parameter efficiency without imposing an explicit low-rank structure on ΔW. To this end, HRHA partitions the weight update into non-overlapping blocks and parameterizes each block using shared spectral bases and block-specific coefficients, as illustrated in Fig. 2.

Block-wise Decomposition. Consider a weight update $\Delta W \in \mathbb{R}^{m\times n}$. We partition it into non-overlapping blocks $B_{ij} \in \mathbb{R}^{b\times b}$:

$$\Delta W = \begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \cdots & w_{mn} \end{bmatrix} = \begin{bmatrix} B_{11} & \cdots & B_{1n'} \\ \vdots & \ddots & \vdots \\ B_{m'1} & \cdots & B_{m'n'} \end{bmatrix}, \quad B_{ij} = \begin{bmatrix} w_{s_1 s_2} & \cdots & w_{s_1 e_2} \\ \vdots & \ddots & \vdots \\ w_{e_1 s_2} & \cdots & w_{e_1 e_2} \end{bmatrix}, \tag{1}$$

where $m' = m/b$ and $n' = n/b$, $s_1 = (i-1)b + 1$, $s_2 = (j-1)b + 1$, $e_1 = ib$, and $e_2 = jb$. This block-wise decomposition transforms the learning of one large weight update into the learning of multiple localized components, allowing different regions of the weight matrix to receive distinct adaptation coefficients.

SVD-inspired Parameterization. Each block can be represented using SVD: $B_{ij} = U_{ij}\Sigma_{ij}V_{ij}^\top$, where $U_{ij}, V_{ij} \in \mathbb{R}^{b\times b}$ are orthogonal matrices and $\Sigma_{ij} \in \mathbb{R}^{b\times b}$ is diagonal. Substituting these decompositions into the block structure yields

$$\Delta W = \begin{bmatrix} U_{11}\Sigma_{11}V_{11}^\top & \cdots & U_{1n'}\Sigma_{1n'}V_{1n'}^\top \\ \vdots & \ddots & \vdots \\ U_{m'1}\Sigma_{m'1}V_{m'1}^\top & \cdots & U_{m'n'}\Sigma_{m'n'}V_{m'n'}^\top \end{bmatrix}. \tag{2}$$

Although learning independent spectral bases for every block would provide maximum flexibility, doing so would introduce a large number of additional parameters. HRHA therefore shares the spectral bases across all blocks within the same layer: $U_{ij} = U, V_{ij} = V, \forall(i,j)$. Therefore,

$$\Delta W = \begin{bmatrix} U\Sigma_{11}V^\top & \cdots & U\Sigma_{1n'}V^\top \\ \vdots & \ddots & \vdots \\ U\Sigma_{m'1}V^\top & \cdots & U\Sigma_{m'n'}V^\top \end{bmatrix} = (I_{m'} \otimes U)\,\tilde{\Sigma}\,(I_{n'} \otimes V)^\top, \tag{3}$$

where $\otimes$ denotes the Kronecker product and $\tilde{\Sigma} \in \mathbb{R}^{m \times n}$ is the block matrix formed by concatenating all Σ_{ij}:

$$\tilde{\Sigma} = \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1n'} \\ \vdots & \ddots & \vdots \\ \Sigma_{m'1} & \cdots & \Sigma_{m'n'} \end{bmatrix}. \tag{4}$$

Here, each Σ_{ij} is diagonal, and hence the trainable degrees of freedom are given by the diagonal entries of all block coefficients.

To achieve parameter efficiency, HRHA initializes U and V as orthogonal matrices and keeps them fixed throughout fine-tuning. Only the block-specific diagonal coefficients Σ_{ij} are optimized. Thus, HRHA separates the roles of spectral representation and adaptation: the shared bases provide a compact common coordinate system, while the block-specific coefficients determine how each region of the weight matrix adapts to the downstream task. Importantly, parameter efficiency is obtained through *structured sharing* rather than by explicitly restricting the rank of ΔW.

2.2 Rank Properties

A central objective of HRHA is to remove the explicit low-rank bottleneck inherent to conventional low-rank adapters. Although the trainable coefficients are highly structured, the resulting update is not parameterized as a product of two low-dimensional factors. We therefore investigate whether this structure introduces an implicit rank limitation.

Theorem 1 (Rank Freedom of HRHA)**.** *Let $\Delta W \in \mathbb{R}^{m \times n}$ be parameterized by HRHA with block size b, shared orthogonal matrices U and V, and diagonal block coefficients Σ_{ij}. Then HRHA does not impose a prescribed low-rank upper bound on $\mathrm{rank}(\Delta W)$ beyond the intrinsic dimensional bound*

$$\mathrm{rank}(\Delta W) \leq \min(m, n). \tag{5}$$

Moreover, the maximum rank $\min(m, n)$ is attainable by a suitable choice of the block coefficients. Hence, HRHA is rank-unconstrained in the sense that it can represent a full-rank update whenever the matrix dimensions permit it.

Proof. The proof is provided in Section C.1. □

This result distinguishes HRHA from conventional low-rank parameterizations, whose update rank is explicitly bounded by the selected rank parameter. In HRHA, the parameterization determines how the update is structured, but does not prescribe a low-dimensional column or row space for the reconstructed matrix. Consequently, the rank of the learned update is determined by the learned block coefficients rather than by a manually specified low-rank budget.

Empirical rank analysis. We further examine the spectral characteristics of HRHA and compare it with LoRA, LoHa, HiRA, GraLoRA, PoLAR, SVFT, and full fine-tuning (FFT) on Qwen2.5-0.5B using numerical rank, stable rank, and effective rank, as defined in Section B.1. The summary is shown in Fig. 1, while the complete distributions are reported in Fig. 3. HRHA exhibits a consistently high-rank spectral profile across the examined modules. Its numerical rank approaches the maximum attainable value on the large Up and Down matrices, reaching 896, and remains substantially higher than conventional low-rank variants on Q, K, and V. In contrast, LoRA, LoHa, HiRA, GraLoRA, and PoLAR remain strongly constrained by their respective rank budgets. HRHA also achieves substantially higher stable ranks on K, V, Up, and Down, and is second only to SVFT on Q, indicating that its spectral energy is distributed across a broader set of singular directions. Similarly, HRHA attains high effective ranks across most modules, demonstrating that its expressive capacity is not solely attributable to a small number of dominant singular values. Overall, these results show that HRHA simultaneously achieves high numerical rank and substantially higher stable and effective ranks than representative low-rank and high-rank adaptation methods. Thus, the theoretical rank freedom of HRHA translates into an empirically broad utilization of the available spectral degrees of freedom, rather than merely producing additional small singular values.

2.3 Task- and Model-Agnostic Adaptation

The required structure of task-specific weight updates can vary across tasks, models, and layers. In low-rank adaptation, parameter efficiency is achieved by explicitly restricting the update rank, making the adaptation space dependent on a predefined rank budget. Such a structural constraint may be suboptimal when different adaptation settings require updates with different rank or spectral characteristics. HRHA instead achieves parameter efficiency through structured parameter sharing rather than rank restriction. The shared bases U and V provide a compact representation, while the block-specific coefficients Σ_{ij} determine the update for each layer. Since the rank is not explicitly constrained and can attain the maximum possible rank, HRHA reduces the dependence of the adaptation space on task- and model-specific rank choices. This provides a unified parameterization that can accommodate diverse adaptation requirements across tasks and model architectures.

2.4 Optimization Properties of HRHA

In addition to its rank properties, HRHA admits a favorable optimization structure when the adapted layer is trained under a squared loss with the backbone parameters and the shared matrices U and V fixed. In this setting, the reconstructed fine-tuning weight is a linear function of the trainable parameter $\tilde{\Sigma}$. Consequently, the optimization problem induced by HRHA is a convex quadratic program, which enables us to establish global convergence of gradient-based optimization.

For clarity, we consider a single linear layer with input $X \in \mathbb{R}^{n\times d}$ and output targets $Y \in \mathbb{R}^{m\times d}$. Let $W_0 \in \mathbb{R}^{m\times n}$ denote the frozen pre-trained weight matrix and let ΔW denote the trainable weight update. The model output is given by $\hat{Y} = (W_0 + \Delta W)X$.

Theorem 2 (Linear Parameterization). *The HRHA weight update $\Delta W = (I_{m'} \otimes U)\tilde{\Sigma}(I_{n'} \otimes V)^\top$ is a linear function of the trainable parameter $\tilde{\Sigma}$. In particular, for any $\tilde{\Sigma}_1, \tilde{\Sigma}_2 \in \mathbb{R}^{m\times n}$ and any scalars $a, c \in \mathbb{R}$, there holds $\Delta W(a\tilde{\Sigma}_1 + c\tilde{\Sigma}_2) = a\Delta W(\tilde{\Sigma}_1) + c\Delta W(\tilde{\Sigma}_2)$.*

Proof. Let $L = I_{m'} \otimes U, R = (I_{n'} \otimes V)^\top$. For any $\tilde{\Sigma}_1, \tilde{\Sigma}_2 \in \mathbb{R}^{m\times n}$ and $a, c \in \mathbb{R}$, we have

$$\Delta W(a\tilde{\Sigma}_1 + c\tilde{\Sigma}_2) = L(a\tilde{\Sigma}_1 + c\tilde{\Sigma}_2)R = aL\tilde{\Sigma}_1 R + cL\tilde{\Sigma}_2 R = a\Delta W(\tilde{\Sigma}_1) + c\Delta W(\tilde{\Sigma}_2). \quad (6)$$

Therefore, the mapping from $\tilde{\Sigma}$ to ΔW is linear. □

Theorem 3 (Convexity). *Consider a linear layer with the squared loss $\mathcal{L}(\tilde{\Sigma}) = \frac{1}{2}\left\|(W_0 + L\tilde{\Sigma}R)X - Y\right\|_F^2$. Then $\mathcal{L}(\tilde{\Sigma})$ is a convex quadratic function of $\tilde{\Sigma}$. More precisely, letting $\tilde{\sigma} = \mathrm{vec}(\tilde{\Sigma})$, the objective can be written as $\mathcal{L}(\tilde{\sigma}) = \frac{1}{2}\|A\tilde{\sigma} - y\|_2^2$, where $A = X^\top R^\top \otimes L, y = \mathrm{vec}(Y - W_0X)$. Consequently, $\nabla_{\tilde{\sigma}}^2\mathcal{L} = A^\top A \succeq 0$, and hence $\mathcal{L}$ is convex.*

Proof. The proof is available in Section C.3. □

Theorem 4 (Global Convergence). *Following the setting of Theorem 3, let $\mathcal{L}(\tilde{\sigma}) = \frac{1}{2}\|A\tilde{\sigma} - y\|_2^2$, and denote $H = A^\top A$. Let $\tilde{\sigma}^{(t)}$ be generated by gradient descent: $\tilde{\sigma}^{(t+1)} = \tilde{\sigma}^{(t)} - \eta\nabla\mathcal{L}(\tilde{\sigma}^{(t)})$, where $0 < \eta < \frac{2}{\lambda_{\max}(H)}$. Then: 1) $\mathcal{L}(\tilde{\sigma}^{(t)})$ converges monotonically to the global minimum $\mathcal{L}^\star$. 2) The iterates $\tilde{\sigma}^{(t)}$ converge to a global minimizer of $\mathcal{L}$. 3) If A has full column rank, equivalently $H \succ 0$, the global minimizer is unique, and the convergence is linear: $\|\tilde{\sigma}^{(t)} - \tilde{\sigma}^\star\|_2 \le \rho^t\|\tilde{\sigma}^{(0)} - \tilde{\sigma}^\star\|_2$, where $\rho = \max_i |1 - \eta\lambda_i(H)| < 1$.*

Proof. The proof is available in Section C.4. □

Summary. The linear parameterization establishes that HRHA does not sacrifice the representational capacity of the fine-tuning weight space; the convexity and global-convergence results demonstrate that this expressive parameterization can also be optimized in a theoretically well-behaved manner. Together, these properties establish that HRHA simultaneously provides *strong expressive capacity and favorable optimization geometry*.

Table 1: Results averaged over 8 commonsense reasoning tasks. SVFT on Qwen2.5-7B is missing due to GPU memory limitations.

Methods	Qwen2.5-0.5B		Qwen2.5-3B		Phi3-3B		Llama3.2-3B		Qwen2.5-7B	
	15K	170K	15K	170K	15K	170K	15K	170K	15K	170K
LoRA(r=16)	54.10	58.62	78.98	82.71	79.64	83.30	66.77	78.77	84.32	87.09
LoRA(r=32)	51.56	**59.61**	79.24	82.84	80.77	83.98	68.12	**79.55**	84.67	**87.45**
LoHa(r=16)	42.63	48.75	72.11	76.88	74.83	79.49	53.56	68.36	80.61	84.72
LoHa(r=32)	42.39	51.80	69.56	79.32	76.35	79.73	55.05	69.95	80.88	85.27
HiRA(r=16)	31.39	36.83	67.33	67.97	70.97	73.99	9.28	40.12	77.73	78.17
HiRA(r=32)	31.90	37.46	67.53	70.12	71.10	74.39	8.99	42.23	77.77	77.84
GraLoRA(r=16)	34.28	53.18	72.13	78.14	75.16	80.38	50.61	66.42	80.13	84.23
GraLoRA(r=32)	37.95	47.16	71.75	77.12	75.56	79.31	50.42	64.27	79.78	83.85
PoLAR(r=16)	51.03	59.65	73.89	73.36	73.21	71.23	34.74	62.59	79.39	81.52
PoLAR(r=32)	52.54	57.04	74.56	80.29	77.15	79.05	62.07	70.95	81.43	83.30
SVFT	36.39	38.65	67.20	71.12	72.13	74.99	16.74	45.14	/	/
HRHA	**56.14**	59.09	**79.62**	**82.86**	**82.68**	**84.42**	**71.85**	78.93	**85.43**	86.31

3 EXPERIMENTS

In this section, we present comprehensive comparative experiments to evaluate the effectiveness of HRHA across five representative benchmarks, and the ablation studies on block size and initialization of U, V. The detailed experiment settings and analysis are provided in Sec. D.

3.1 MAIN RESUTS

Commonsense Reasoning Table 1 summarizes the average accuracy over eight commonsense reasoning benchmarks, and Table 8 lists the corresponding number of trainable parameters. HRHA exhibits a clear advantage in the low-resource setting (15K), where it achieves the highest accuracy on all five evaluated models, with particularly large gains over the best LoRA variant (up to 4.58 points on Qwen2.5-0.5B). Under full-data training (170K), HRHA remains competitive, attaining the best results on Qwen2.5-3B and Phi3-3B and staying within 0.52–1.14 points of the best on the remaining models, while using markedly fewer parameters. In contrast, higher-rank methods including LoHa, HiRA, GraLoRA, and PoLAR show inconsistent or even degraded performance, indicating that simply increasing rank does not ensure robust commonsense reasoning. These results demonstrate that HRHA, by removing the low-rank constraint while retaining structured parameter sharing, offers an effective balance between data efficiency, parameter efficiency, and task performance across diverse model scales.

Natural Language Understanding Table 2 reports GLUE results for $\text{RoBERTa}_{\text{base}}$ and $\text{RoBERTa}_{\text{large}}$ with all baselines using rank 16. HRHA achieves the best average score on both scales (82.6 and 86.5) while using the fewest trainable parameters (0.33M and 1.18M), substantially less than the 2.65M/7.07M of LoRA and other rank-based methods. The gains are most pronounced on challenging tasks such as CoLA and RTE, where HRHA outperforms the strongest baselines by clear margins. In contrast, higher-rank variants including LoHa, PoLAR, and HiRA either underperform or exhibit severe training instability, indicating that merely increasing rank does not ensure robust adaptation. These results demonstrate that HRHA effectively balances expressive capacity and parameter efficiency.

Image Classification Table 3 reports image classification results on ViT-Base and ViT-Large. HRHA achieves an average accuracy of 87.66% on ViT-Base, essentially matching FFT (87.74%), and 90.23% on ViT-Large, ranking second only to HiRA (91.54%), while using only 0.26M and 0.59M trainable parameters, respectively. It uses the fewest parameters among all PEFT methods and delivers the largest gains on challenging datasets such as StanfordCars and FGVC, where it outperforms the strongest baselines by wide margins. In contrast, LoHa, GraLoRA, and PoLAR consistently underperform, and HiRA and SVFT require substantially more parameters for compa-

Table 2: Results with RoBERTa$_{\text{base}}$ and RoBERTa$_{\text{large}}$ on the GLUE benchmark. The best results in each group are shown in bold. We report Matthew's correlation for CoLA, Pearson correlation for STS-B, and accuracy for all other datasets. We report the median performance trained on 10 epochs from 5 runs using different random seeds. The rank of baselines is set to 16.

	Method	#Params	SST-2	MRPC	CoLA	QNLI	RTE	STS-B	Avg.
BASE	LoRA	2.65M	**94.2**$_{\pm 0.5}$	86.7$_{\pm 0.2}$	58.8$_{\pm 0.7}$	**92.5** $_{\pm 0.1}$	**72.6**$_{\pm 1.4}$	90.3$_{\pm 0.1}$	82.5$_{\pm 0.2}$
	LoHa	2.65M	94.2$_{\pm 0.4}$	70.3$_{\pm 0.4}$	47.6$_{\pm 0.6}$	91.9$_{\pm 0.2}$	56.8$_{\pm 0.6}$	86.1$_{\pm 0.1}$	74.5$_{\pm 0.2}$
	HiRA	2.65M	92.5$_{\pm 0.3}$	68.4$_{\pm 0.0}$	0.0$_{\pm 0.0}$	89.7$_{\pm 0.3}$	47.9$_{\pm 2.1}$	14.2$_{\pm 3.9}$	52.1$_{\pm 0.9}$
	GraLoRA	2.65M	94.2$_{\pm 0.3}$	87.4$_{\pm 0.5}$	58.0$_{\pm 1.1}$	92.5$_{\pm 0.2}$	71.1$_{\pm 1.4}$	90.1$_{\pm 0.1}$	82.2$_{\pm 0.2}$
	PoLAR	2.65M	93.4$_{\pm 0.3}$	80.1$_{\pm 1.4}$	45.7$_{\pm 3.1}$	91.3$_{\pm 0.3}$	61.2$_{\pm 2.5}$	87.1$_{\pm 0.6}$	76.5$_{\pm 0.6}$
	SVFT	0.93M	93.9$_{\pm 0.2}$	79.8$_{\pm 1.0}$	49.6$_{\pm 1.0}$	91.1$_{\pm 0.2}$	61.2$_{\pm 1.7}$	87.2$_{\pm 0.1}$	77.1$_{\pm 0.5}$
	HRHA	0.33M	94.1$_{\pm 0.2}$	**87.7**$_{\pm 0.9}$	**59.7**$_{\pm 1.2}$	92.3$_{\pm 0.3}$	71.5$_{\pm 0.9}$	**90.3**$_{\pm 0.1}$	**82.6**$_{\pm 0.3}$
LARGE	LoRA	7.07M	95.5$_{\pm 0.3}$	**88.7**$_{\pm 0.5}$	63.8$_{\pm 1.3}$	**94.3**$_{\pm 0.2}$	82.2$_{\pm 1.5}$	**91.9**$_{\pm 0.2}$	86.1$_{\pm 0.4}$
	LoHa	7.07M	95.8$_{\pm 0.1}$	70.9$_{\pm 0.4}$	51.7$_{\pm 1.2}$	94.2$_{\pm 0.2}$	54.5$_{\pm 0.8}$	82.4$_{\pm 0.4}$	74.9$_{\pm 0.3}$
	HiRA	7.07M	95.3$_{\pm 0.3}$	68.4$_{\pm 0.0}$	0.0$_{\pm 0.0}$	93.0$_{\pm 0.2}$	48.2$_{\pm 2.1}$	0.3$_{\pm 2.4}$	50.8$_{\pm 0.7}$
	GraLoRA	7.07M	95.6$_{\pm 0.3}$	88.3$_{\pm 0.6}$	65.0$_{\pm 0.7}$	94.3 $_{\pm 0.2}$	82.0$_{\pm 2.1}$	91.9$_{\pm 0.2}$	86.2$_{\pm 0.5}$
	PoLAR	7.07M	95.5$_{\pm 0.2}$	76.6$_{\pm 2.9}$	53.4$_{\pm 1.1}$	94.0$_{\pm 0.2}$	58.5$_{\pm 2.1}$	86.6$_{\pm 0.6}$	77.4$_{\pm 0.5}$
	SVFT	2.50M	**95.9**$_{\pm 0.2}$	81.3$_{\pm 1.2}$	55.5$_{\pm 0.5}$	93.9$_{\pm 0.3}$	62.3$_{\pm 0.7}$	86.9$_{\pm 0.6}$	79.3$_{\pm 0.3}$
	HRHA	1.18M	95.6$_{\pm 0.3}$	88.2$_{\pm 1.0}$	**65.1**$_{\pm 0.3}$	94.0$_{\pm 0.2}$	**84.0**$_{\pm 1.8}$	91.8$_{\pm 0.2}$	**86.5**$_{\pm 0.5}$

Table 3: The performance of different fine-tuning methods on various image classification datasets for the ViT-Base and ViT-Large models, averaged over 5 random seeds. The rank of baselines is set to 16. The results of FFT are derived from Zheng et al. (2026). Each PEFT method is trained with its optimal learning rate.

Model	Method	#Param	Accuracy (%)							
			CIFAR10	CIFAR100	Pets	Cars	FGVC	EuroSAT	RESISC45	Avg.
ViT-Base	FFT	85.8M	93.14$_{\pm 0.4}$	79.78$_{\pm 1.2}$	**98.92**$_{\pm 0.1}$	**99.05**$_{\pm 0.1}$	54.84$_{\pm 1.2}$	96.13$_{\pm 0.1}$	92.38$_{\pm 0.1}$	**87.74**
	LoRA	0.74M	98.74$_{\pm 0.08}$	91.81$_{\pm 0.05}$	92.69$_{\pm 0.79}$	39.72$_{\pm 3.62}$	31.14$_{\pm 1.95}$	98.66$_{\pm 0.27}$	94.82$_{\pm 0.17}$	78.22
	LoHa	0.74M	98.42$_{\pm 0.05}$	91.28$_{\pm 0.10}$	92.94$_{\pm 0.36}$	59.82$_{\pm 0.40}$	35.42$_{\pm 0.70}$	97.27$_{\pm 0.11}$	90.32$_{\pm 0.21}$	80.78
	HiRA	0.74M	98.86$_{\pm 0.02}$	**92.05**$_{\pm 0.12}$	93.01$_{\pm 0.79}$	63.17$_{\pm 0.84}$	51.23$_{\pm 1.52}$	98.60$_{\pm 0.43}$	94.62$_{\pm 0.30}$	84.51
	GraLoRA	0.74M	98.58$_{\pm 0.16}$	91.65$_{\pm 0.20}$	93.05$_{\pm 0.38}$	41.47$_{\pm 1.96}$	32.43$_{\pm 1.65}$	**98.73**$_{\pm 0.12}$	**94.82**$_{\pm 0.17}$	78.68
	PoLAR	0.75M	98.80$_{\pm 0.07}$	91.91$_{\pm 0.10}$	93.12$_{\pm 0.33}$	28.75$_{\pm 0.95}$	20.60$_{\pm 1.14}$	98.68$_{\pm 0.11}$	94.01$_{\pm 0.13}$	75.12
	SVFT	0.46M	98.71$_{\pm 0.05}$	91.57$_{\pm 0.12}$	92.77$_{\pm 0.68}$	58.31$_{\pm 0.79}$	48.62$_{\pm 0.70}$	98.71$_{\pm 0.17}$	94.22$_{\pm 0.21}$	83.27
	HRHA	0.26M	**98.88**$_{\pm 0.04}$	91.79$_{\pm 0.20}$	93.68$_{\pm 0.41}$	79.73$_{\pm 0.50}$	**56.98**$_{\pm 0.60}$	98.49$_{\pm 0.17}$	94.09$_{\pm 0.20}$	87.66
ViT-Large	FFT	303.3M	94.43$_{\pm 0.6}$	88.90$_{\pm 0.3}$	**99.15**$_{\pm 0.1}$	**99.04**$_{\pm 0.1}$	68.25$_{\pm 1.6}$	96.43$_{\pm 0.3}$	93.58$_{\pm 0.1}$	91.40
	LoRA	1.77M	98.96$_{\pm 0.20}$	93.36$_{\pm 0.08}$	94.41$_{\pm 0.15}$	82.25$_{\pm 0.29}$	60.61$_{\pm 1.74}$	98.76$_{\pm 0.13}$	95.57$_{\pm 0.22}$	89.13
	LoHa	1.77M	98.96$_{\pm 0.06}$	92.77$_{\pm 0.15}$	92.82$_{\pm 0.52}$	75.39$_{\pm 0.38}$	45.30$_{\pm 0.65}$	97.78$_{\pm 0.11}$	92.38$_{\pm 0.11}$	85.06
	HiRA	1.77M	99.07$_{\pm 0.09}$	**93.71**$_{\pm 0.15}$	93.88$_{\pm 0.58}$	88.29$_{\pm 0.43}$	**71.34**$_{\pm 1.00}$	98.65$_{\pm 0.28}$	**95.87**$_{\pm 0.14}$	**91.54**
	GraLoRA	1.77M	99.02$_{\pm 0.06}$	93.10$_{\pm 0.19}$	94.37$_{\pm 0.24}$	84.87$_{\pm 0.47}$	64.96$_{\pm 1.19}$	**98.98**$_{\pm 0.16}$	95.64$_{\pm 0.06}$	90.13
	PoLAR	1.79M	99.06$_{\pm 0.04}$	92.97$_{\pm 0.25}$	94.12$_{\pm 0.90}$	69.76$_{\pm 4.82}$	54.69$_{\pm 2.04}$	98.74$_{\pm 0.07}$	95.07$_{\pm 0.22}$	86.34
	SVFT	1.03M	**99.13**$_{\pm 0.05}$	93.55$_{\pm 0.11}$	94.16$_{\pm 0.58}$	87.05$_{\pm 0.31}$	67.37$_{\pm 6.38}$	98.87$_{\pm 0.08}$	95.44$_{\pm 0.23}$	90.80
	HRHA	0.59M	99.06$_{\pm 0.05}$	93.68$_{\pm 0.07}$	94.65$_{\pm 0.25}$	83.95$_{\pm 1.13}$	66.25$_{\pm 1.74}$	98.97$_{\pm 0.09}$	95.08$_{\pm 0.16}$	90.23

rable or inferior results. These results demonstrate that HRHA achieves near-FFT accuracy with exceptional parameter efficiency, validating the effectiveness of its parameterization.

Personalized Dialogue Generation Table 4 presents results on personalized dialogue generation. HRHA achieves the highest average on Llama-2-7B (47.92) and ranks competitively on Llama-3-8B (47.39 vs. HiRA's 47.54 and LoRA's 47.27), while using only 4.19M/2.62M parameters—about half or less than the rank-based baselines. Notably, HRHA obtains the best BLEU and METEOR scores on both models, indicating superior generation quality. In contrast, PoLAR collapses completely, and other higher-rank methods show inconsistent performance across models. These results demonstrate that HRHA provides robust, high-quality dialogue generation with strong parameter efficiency.

Arithmetic Reasoning Table 5 presents arithmetic reasoning results on Llama-2-7B and Llama-3-8B. HRHA achieves average accuracies of 63.82 and 75.35, while using only 4.19M/2.62M parameters—about half or less than the rank-based baselines. It obtains the highest scores on MultiArith and AddSub (Llama-2-7B) and MultiArith and SingleEq (Llama-3-8B), demonstrating strong symbolic reasoning ability. In contrast, PoLAR and HiRA show significant performance fluctuations across

Table 4: Results of each method on personalized dialogue generation.

	Method	#Params	BLEU	BERT F1	BERT-R	BERT-P	METEOR	R-L	Average
Llama-2-7B	LoRA	8.38M	2.57	86.14	86.20	86.11	13.21	12.71	47.82
	LoHa	8.38M	**2.59**	86.31	86.34	**86.31**	13.09	12.75	47.90
	HiRA	8.38M	2.29	86.20	86.15	86.29	12.42	12.42	47.63
	GraLoRA	8.38M	2.57	86.02	86.21	85.88	**13.70**	12.84	47.87
	PoLAR	8.40M	0.01	77.30	80.22	74.82	0.16	0.31	38.80
	SVFT	4.45M	2.38	86.27	86.32	86.25	13.08	12.75	47.84
	HRHA	4.19M	2.39	**86.32**	**86.38**	86.28	13.23	**12.93**	**47.92**
Llama-3-8B	LoRA	6.82M	3.40	85.87	85.14	86.66	8.49	14.06	47.27
	LoHa	6.82M	3.13	85.98	85.16	86.86	8.40	13.97	47.25
	HiRA	6.82M	3.26	**86.26**	**85.30**	**87.28**	8.62	**14.50**	**47.54**
	GraLoRA	6.82M	3.31	85.70	85.00	86.45	8.45	13.56	47.08
	PoLAR	6.83M	0.00	76.37	80.07	73.01	0.08	0.23	38.29
	SVFT	2.78M	2.90	86.01	85.16	86.90	8.30	13.81	47.18
	HRHA	2.62M	**3.47**	85.92	85.18	86.72	**8.90**	14.19	47.39

Table 5: Results of each method on arithmetic reasoning.

	Method	#Params	MultiArith	gsm8k	AddSub	AQuA	SingleEq	SVAMP	Average
Llama-2-7B	LoRA	8.39M	91.83	**36.16**	87.59	19.29	**89.96**	53.30	63.02
	LoHa	8.39M	79.50	28.20	80.76	19.69	80.12	48.50	56.13
	HiRA	8.39M	60.67	21.08	69.11	18.11	76.18	40.90	47.67
	GraLoRA	8.39M	91.50	35.63	89.62	**23.62**	88.19	**54.70**	**63.88**
	PoLAR	8.40M	85.83	30.17	82.53	22.44	82.87	51.60	59.24
	SVFT	4.45M	65.67	27.75	82.28	20.08	81.30	50.50	54.60
	HRHA	4.19M	**92.67**	34.57	**89.87**	23.23	87.99	54.60	63.82
Llama-3-8B	LoRA	6.82M	94.67	61.03	90.13	28.35	94.09	75.90	74.03
	LoHa	6.82M	92.17	62.24	90.38	27.95	93.31	73.80	73.31
	HiRA	6.82M	86.67	53.68	90.89	**31.50**	91.14	71.20	70.84
	GraLoRA	6.82M	95.67	61.56	89.11	25.98	94.49	75.40	73.70
	PoLAR	6.83M	95.83	**64.29**	**91.90**	29.53	94.69	**76.70**	**75.49**
	SVFT	2.78M	87.67	55.57	88.61	26.77	92.13	69.80	70.09
	HRHA	2.62M	**97.83**	63.68	89.37	29.53	**95.28**	76.40	75.35

models, and GraLoRA requires twice the parameters for similar results. These findings confirm that HRHA consistently combines high accuracy with excellent parameter efficiency on arithmetic reasoning tasks.

3.2 Cross-Task and Cross-Model Analysis

Across all five benchmarks, HRHA demonstrates consistent effectiveness across substantially different adaptation settings. The evaluated tasks span commonsense reasoning, natural language understanding, image classification, personalized dialogue generation, and arithmetic reasoning, covering both language and vision as well as discriminative and generative objectives. At the model level, HRHA is evaluated on Qwen2.5, Phi-3, Llama-2, Llama-3, RoBERTa, and ViT, ranging from sub-billion to multi-billion parameter models and across different model architectures. Despite these differences, HRHA uses the same block-wise parameterization without introducing task- or model-specific rank choices.

The results further show that this consistency is not limited to a particular data regime or model scale. HRHA performs particularly strongly in the low-resource commonsense reasoning setting, while remaining competitive under full-data training. It also maintains strong parameter efficiency across both small and large models, consistently requiring substantially fewer trainable parameters than rank-based baselines. These observations are consistent with the design of HRHA: structured parameter sharing provides parameter efficiency, while the absence of an explicit low-rank constraint

Table 6: The ablation studies of HRHA for the block size on image classification datasets.

Block Size	#Param	Accuracy (%)							
		OxfordPets	StanfordCars	CIFAR10	EuroSAT	FGVC	RESISC45	CIFAR100	Avg.
256*256	196K	$93.95_{\pm 0.2}$	$81.75_{\pm 0.3}$	$99.07_{\pm 0.1}$	$98.77_{\pm 0.2}$	$56.55_{\pm 1.0}$	$94.56_{\pm 0.2}$	$93.32_{\pm 0.1}$	88.28
128*128	393K	$94.19_{\pm 0.4}$	$83.34_{\pm 0.4}$	$99.06_{\pm 0.4}$	$98.97_{\pm 0.1}$	$62.90_{\pm 1.2}$	$98.97_{\pm 0.1}$	$93.49_{\pm 0.2}$	89.58
64*64	786K	$94.61_{\pm 0.5}$	$84.93_{\pm 0.4}$	$99.08_{\pm 0.1}$	$99.04_{\pm 0.2}$	$67.85_{\pm 0.9}$	$95.17_{\pm 0.3}$	$93.59_{\pm 0.2}$	90.61
32*32	1572K	$94.85_{\pm 0.3}$	$85.98_{\pm 0.3}$	$99.12_{\pm 0.1}$	$98.97_{\pm 0.1}$	$70.12_{\pm 1.0}$	$95.15_{\pm 0.1}$	$93.51_{\pm 0.2}$	91.1

Table 7: The ablation studies of HRHA for the U, V initialization on GLUE.

	Method	SST-2	MRPC	CoLA	QNLI	RTE	STS-B	Avg.
BASE	Joint bi-sided PCA	$94.1_{\pm 0.2}$	$87.7_{\pm 0.9}$	$59.7_{\pm 1.2}$	$92.3_{\pm 0.3}$	$71.5_{\pm 0.9}$	$90.3_{\pm 0.1}$	$82.6_{\pm 0.3}$
	Random Orthogonal	$94.1_{\pm 0.2}$	$87.6_{\pm 0.5}$	$59.3_{\pm 0.8}$	$92.5_{\pm 0.5}$	$72.1_{\pm 1.3}$	$90.2_{\pm 0.1}$	$82.6_{\pm 0.2}$
	Uniform	$82.6_{\pm 0.7}$	$71.0_{\pm 0.3}$	$26.7_{\pm 3.1}$	$70.5_{\pm 1.5}$	$56.0_{\pm 3.1}$	$60.1_{\pm 0.7}$	$61.1_{\pm 1.1}$
LARGE	Joint bi-sided PCA	$95.6_{\pm 0.3}$	$88.2_{\pm 1.0}$	$65.1_{\pm 0.3}$	$94.0_{\pm 0.2}$	$84.0_{\pm 1.8}$	$91.8_{\pm 0.2}$	$86.5_{\pm 0.5}$
	Random Orthogonal	$95.6_{\pm 0.3}$	$88.5_{\pm 0.7}$	$66.0_{\pm 2.6}$	$94.3_{\pm 0.2}$	$84.3_{\pm 1.2}$	$91.6_{\pm 0.2}$	$86.7_{\pm 0.3}$
	Uniform	$84.4_{\pm 1.1}$	$70.8_{\pm 0.8}$	$9.2_{\pm 6.4}$	$65.5_{\pm 2.3}$	$51.6_{\pm 2.3}$	$45.0_{\pm 1.1}$	$54.4_{\pm 1.0}$

allows the effective rank and spectral structure of the update to adapt to the requirements of different tasks and models. Thus, rather than relying on a task- or model-specific rank assumption, HRHA provides a unified adaptation mechanism that remains effective across diverse tasks, architectures, and training regimes.

3.3 Ablation Studies

Block Size Table 6 shows the block-size ablation on image classification. Decreasing the block size from 256×256 to 32×32 consistently improves average accuracy from 88.28 to 91.10, but at the cost of doubling the parameter count at each step (196K to 1572K). The gains are largest on challenging datasets such as StanfordCars and FGVC, while nearly saturated tasks like CIFAR10 and EuroSAT benefit little. Since the improvement from 128×128 to 32×32 is only 1.52 points with a $4\times$ parameter increase, we adopt 128×128 as the default, offering the best trade-off between accuracy and parameter efficiency.

Initialization of U and V Table 7 ablates the initialization of U and V on GLUE. Uniform (non-orthogonal) initialization causes severe performance collapse on both RoBERTa$_{\text{base}}$ and RoBERTa$_{\text{large}}$, with average scores dropping from 82.6 to 61.1 and from 86.5 to 54.4, respectively, and CoLA falling to near-random levels. In contrast, the two orthogonal strategies, joint bi-sided PCA and random orthogonal (details provided in Sec. C.5), perform almost identically, with only marginal differences within noise. This confirms that orthogonality of the shared singular matrices is essential for HRHA, while the specific orthogonal bases has negligible impact.

4 Conclusion

In this work, we proposed **HRHA**, a high-rank and high-applicability PEFT method that decouples parameter efficiency from explicit low-rank parameterization. HRHA achieves parameter efficiency and preserves flexible adaptation without imposing an explicit rank constraint on the resulting update. We theoretically established its rank flexibility and linear parameterization. Extensive experiments across diverse tasks, model architectures, and scales demonstrate that HRHA consistently achieves strong performance with substantially fewer trainable parameters.

5 EXTRA MATERIAL

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing, including grammar correction, academic writing refinement, and improvement of the clarity and organization of the manuscript. We also used generative AI tools to assist with code debugging and implementation-related questions during the development of the experimental framework. These tools were used as assistants and did not replace the authors' judgment in research design, theoretical development, experimental methodology, or interpretation of results.

We have not used generative AI tools to generate experimental results, perform experiments in place of the authors, or determine the scientific conclusions of this work. All theoretical formulations, proofs, experimental settings, reported results, and scientific claims were developed, verified, and reviewed by the authors.

All AI-assisted content was carefully reviewed by the authors. In particular, AI-assisted mathematical formulations and theoretical arguments were independently checked for correctness, and AI-assisted code was inspected, executed, and tested by the authors. The authors also verified that the reported experimental results were obtained from the actual experimental runs and that the corresponding claims are supported by the reported evidence. We take full responsibility for the final content of this work, including all text, claims, mathematical formulations, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, private or personally identifiable information, or the collection of new sensitive data. We evaluate our method using publicly available benchmark datasets and pre-trained models, following their respective licenses and intended research use. The experiments focus on parameter-efficient fine-tuning and do not target high-risk or harmful applications.

We acknowledge that pre-trained models and benchmark datasets may contain biases or other limitations inherited from their original training data. Our work does not attempt to eliminate these biases, and the reported results should therefore be interpreted within the scope and limitations of the evaluated models and datasets. We do not release or expose private information through this work.

Finally, we have conducted the experiments and analyses in accordance with standard research practices and report the methodology, experimental settings, and results as accurately as possible. We take responsibility for the integrity and reproducibility of the reported research.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure that the proposed method and experimental results are reproducible. The complete formulation and implementation details of HRHA are described in the main paper, including its block-wise parameterization, shared spectral bases, rank properties, and optimization formulation. The assumptions and complete proofs of the theoretical results are provided in the appendix. Detailed experimental configurations, including datasets, model architectures, training settings, hyperparameters, and evaluation protocols, are documented in the appendix and supplementary materials. The supplementary materials will also include the source code and scripts required to reproduce the reported experiments, together with instructions for environment setup and evaluation. All reported results are obtained from actual experimental runs using the configurations described in the paper and supplementary materials.

REFERENCES

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

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

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems, 2021. *URL https://arxiv. org/abs/2110.14168*, 9, 2021.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. The second conversational intelligence challenge (convai2). In *The NeurIPS'18 Competition: From Machine Learning to Intelligent Conversations*, pp. 187–208. Springer, 2019.

William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the third international workshop on paraphrasing (IWP2005)*, 2005.

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

Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. Parameter-efficient fine-tuning with discrete fourier transform. *arXiv preprint arXiv:2405.03003*, 2024.

Yuxuan Gu, Wuyang Zhou, Giorgos Iacovides, and Danilo Mandic. Tera: Vector-based random tensor network for high-rank adaptation of large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2314–2329, 2026.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 523–533, 2014.

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

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 *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 5254–5276, 2023.

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

Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. Fedpara: Low-rank hadamard product for communication-efficient federated learning. *arXiv preprint arXiv:2108.06098*, 2021.

Yeonjoon Jung, Daehyun Ahn, Hyungjun Kim, Taesu Kim, and Eunhyeok Park. Gralora: Granular low-rank adaptation for parameter-efficient fine-tuning. *Advances in Neural Information Processing Systems*, 38:95822–95845, 2026.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597, 2015.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. Mawps: A math word problem repository. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pp. 1152–1157, 2016.

Yang Li, Shaobo Han, and Shihao Ji. Vb-lora: Extreme parameter efficient fine-tuning with vector banks. *Advances in Neural Information Processing Systems*, 37:16724–16751, 2024.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 158–167, 2017.

Vijay Lingam, Atula Tejaswi, Aditya Vavre, Aneesh Shetty, Gautham K Gudur, Joydeep Ghosh, Alex Dimakis, Eunsol Choi, Aleksandar Bojchevski, and Sujay Sanghavi. Svft: Parameter-efficient fine-tuning with singular vectors. *Advances in Neural Information Processing Systems*, 37:41425–41446, 2024.

Kai Lion, Liang Zhang, Bingcong Li, and Niao He. Polar: Polar-decomposed low-rank adapter representation. *Advances in Neural Information Processing Systems*, 38:149313–149358, 2026.

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

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

Francesco Mezzadri. How to generate random matrices from the classical compact groups. *arXiv preprint math-ph/0609050*, 2006.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 2080–2094, 2021.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

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

Subhro Roy and Dan Roth. Solving general arithmetic word problems. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1743–1752, 2015.

Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM (JACM)*, 54(4):21–es, 2007.

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

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Jingjing Zheng, Wanglong Lu, Yiming Dong, Chaojie Ji, Yankai Cao, and Zhouchen Lin. Adamss: Adaptive multi-subspace approach for parameter-efficient fine-tuning. *Advances in Neural Information Processing Systems*, 38:19746–19774, 2026.

A Related Work

A growing body of work has sought to relax the rank limitation of LoRA while retaining its parameter efficiency. These approaches employ diverse parameterizations to increase the rank or expressive capacity of fine-tuning updates without introducing the full parameter cost of fine-tuning. LoHa Hyeon-Woo et al. (2021) and related Hadamard-product parameterizations construct weight updates as the element-wise product of multiple low-rank matrices, enabling substantially higher-rank updates with a comparable parameter budget. Similarly, HiRA Huang et al. (2025) leverages a Hadamard-product parameterization involving the pretrained weights to increase the attainable rank of the adaptation update. GraLoRA Jung et al. (2026) adopts a granular block-wise parameterization, partitioning the weight matrix into smaller regions and assigning low-rank adapters at a finer granularity, thereby increasing the flexibility of the resulting update. PoLAR Lion et al. (2026) parameterizes the update as $X\Theta Y^\top$, with the left and right factors constrained to Stiefel manifolds, enabling high-rank adaptation through manifold optimization.

Although these methods can substantially increase the *nominal numerical rank* of the learned updates, a high numerical rank does not necessarily imply that the update effectively utilizes a broad range of spectral directions. As illustrated in Fig. 3, many high-rank variants still exhibit relatively low stable and effective ranks, suggesting that their spectral energy remains concentrated in a limited number of dominant directions. This observation highlights an important distinction between *numerical rank* and the effective utilization of the available spectral degrees of freedom: increasing the attainable numerical rank alone does not necessarily lead to a more balanced spectral structure.

Another line of work enhances adaptation capacity through fixed spectral bases or structured linear parameterizations. SVFT Lingam et al. (2024), for example, constructs fine-tuning updates from the singular vectors of the pretrained weight matrix and learns only a selected subset of the corresponding spectral coefficients. Because the spectral bases are fixed, SVFT yields a linear parameterization with respect to its trainable coefficients. Under a single-layer squared-loss formulation, this leads to a convex quadratic optimization problem. However, SVFT employs a global spectral parameterization: the same left and right singular bases are shared across the entire weight matrix, while only a selected subset of spectral coefficients is optimized. This global structure may limit the ability to capture heterogeneous adaptation patterns across different regions of a weight matrix.

HRHA adopts a different structured parameterization that combines shared spectral bases with block-specific adaptation. Specifically, HRHA represents the update as

$$\Delta W = (I_{m'} \otimes U)\,\tilde{\Sigma}\,(I_{n'} \otimes V)^\top,$$

where U and V are fixed shared orthogonal bases and $\tilde{\Sigma}$ contains the trainable diagonal coefficients associated with individual blocks. Unlike global spectral parameterizations, HRHA partitions the weight matrix into non-overlapping blocks, allowing different regions to learn independent coefficients while sharing the same spectral bases. Importantly, the block-specific coefficients are not subject to an explicit low-rank constraint or a prescribed sparse spectral support. Consequently, the resulting update can attain the maximum possible rank, up to $\min(m, n)$, while retaining a substantially smaller trainable parameter budget than full fine-tuning. Moreover, because U and V are fixed, the trainable coefficients enter the update linearly. Under the single-layer squared-loss setting, the resulting optimization problem is therefore a convex quadratic problem, for which standard gradient-based optimization admits well-understood convergence properties.

These distinctions motivate a broader perspective on high-rank PEFT. Existing methods demonstrate that the low-rank constraint of LoRA can be relaxed without restoring the full parameter cost of fine-tuning. However, the attainable numerical rank and the effective utilization of spectral degrees of freedom are not equivalent. Hadamard-product and factorized approaches such as LoHa, HiRA, and GraLoRA can increase numerical rank while still exhibiting concentrated spectral energy. SVFT provides a linear and convex parameterization, but its global spectral structure may limit regional adaptation flexibility. HRHA instead combines *high-rank expressivity*, *block-wise adaptability*, and *structured parameter sharing* within a unified framework, while retaining a linear parameterization and the corresponding convex-quadratic optimization structure under the fixed-basis, single-layer squared-loss setting.

B PRELIMINARIES

In this section, we first introduce three complementary rank metrics—numerical rank, stable rank, and effective rank—to characterize different aspects of the dimensionality and spectral structure of weight updates learned during fine-tuning. Numerical rank measures the number of singular directions with non-negligible magnitude, while stable rank and effective rank quantify complementary aspects of spectral energy dispersion and spectral utilization. Together, these metrics provide a more informative characterization of adaptation capacity than numerical rank alone.

Building on these metrics, we then examine the structural implications of explicit low-rank constraints in PEFT. In particular, we show that a prescribed rank imposes a hard upper bound on the number of available adaptation directions, thereby restricting the subspace in which the weight update can reside. This perspective motivates the use of rank statistics to distinguish nominal rank from the effective utilization of spectral degrees of freedom, and provides the basis for our subsequent analysis of HRHA.

B.1 RANK METRICS

To comprehensively characterize the spectral properties of weight updates learned by different PEFT methods, we consider three complementary rank measures: numerical rank, stable rank, and effective rank. Each metric captures a distinct aspect of the singular-value distribution of the update matrix $\Delta W \in \mathbb{R}^{m\times n}$, and together they provide a multifaceted view of how the adaptation utilizes the available spectral degrees of freedom.

Let the SVD of ΔW be given by $\Delta W = \sum_{i=1}^{\min(m,n)} \sigma_i \, \mathbf{u}_i \mathbf{v}_i^\top$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0$ denote the singular values in non-increasing order.

Numerical rank. The numerical rank, denoted $\mathrm{rank}_\varepsilon(\Delta W)$, counts the number of singular values that exceed a prescribed threshold $\varepsilon > 0$:

$$\mathrm{rank}_\varepsilon(\Delta W) = \big|\{i : \sigma_i > \varepsilon\}\big|.$$

Following HiRA Huang et al. (2025), we set $\varepsilon = 5 \times 10^{-3}$ in our experiments. This threshold provides a consistent criterion for distinguishing non-negligible singular directions from directions with very small singular values. The numerical rank therefore provides an empirical estimate of the number of actively utilized adaptation directions. For a low-rank parameterization satisfying $\mathrm{rank}(\Delta W) \leq r$, the numerical rank cannot exceed r, whereas rank-unconstrained parameterizations can activate a substantially larger number of directions.

Stable rank. The stable rank Rudelson & Vershynin (2007) is defined as the ratio of the squared Frobenius norm to the squared spectral norm:

$$\mathrm{Srank}(\Delta W) = \frac{\|\Delta W\|_F^2}{\|\Delta W\|_2^2} = \frac{\sum_i \sigma_i^2}{\sigma_1^2}.$$

Unlike numerical rank, which depends on a hard threshold, stable rank provides a continuous measure of the dispersion of squared spectral energy. When most of the energy is concentrated in the largest singular value, the stable rank approaches 1. Conversely, a larger stable rank indicates that a broader collection of singular directions contributes substantially to the overall energy of the update. Thus, stable rank captures the extent to which the update utilizes multiple directions beyond its dominant singular mode.

For a matrix of rank at most r, the stable rank is also bounded by r. Importantly, however, numerical rank and stable rank measure different properties: a matrix may have a relatively large numerical rank while still possessing a much smaller stable rank if its singular values decay rapidly. This distinction allows us to assess whether an apparently high-rank update actually distributes substantial spectral energy across multiple directions.

Effective rank. The effective rank Roy & Vetterli (2007) is derived from the entropy of the normalized singular value distribution. Let $p_i = \sigma_i / \sum_j \sigma_j$ be the proportion of the total spectral mass

carried by the i-th singular value. The Shannon entropy of this distribution is $H = -\sum_i p_i \log p_i$, and the effective rank is defined as

$$\mathrm{Erank}(\Delta W) = \exp(H).$$

The effective rank reaches its maximum when the non-zero singular values are uniformly distributed and decreases as the spectral mass becomes increasingly concentrated in a small number of singular directions. It therefore provides a complementary measure of the balance of the singular-value distribution. In particular, two updates with the same numerical rank can have substantially different effective ranks, reflecting different degrees of spectral concentration.

Taken together, the three metrics characterize complementary properties of the learned update: numerical rank measures the number of non-negligible singular directions, stable rank measures the dispersion of squared spectral energy, and effective rank measures the balance of the singular-value distribution. This combination enables us to distinguish the *nominal dimensionality* of an update from the extent to which its available spectral degrees of freedom are effectively utilized.

B.2 Limitations of Low-Rank Adaptation

We next examine the structural limitations introduced by explicit low-rank parameterization from the perspectives of numerical rank, stable rank, and effective rank. Consider the standard LoRA formulation

$$\Delta W = BA, \qquad B \in \mathbb{R}^{m\times r}, \quad A \in \mathbb{R}^{r\times n},$$

where $r \ll \min(m, n)$. By construction,

$$\mathrm{rank}(\Delta W) \le r.$$

Thus, low-rank adaptation restricts the update to a matrix subspace whose dimension is fundamentally limited by the prescribed rank. This rank ceiling has direct consequences for the dimensionality of the adaptation space and, more generally, determines how many independent singular directions can be represented.

Restricted numerical rank. The most direct consequence of the low-rank constraint is the hard upper bound

$$\mathrm{rank}(\Delta W) \le r \ll \min(m, n).$$

Consequently, at most r singular directions can have non-zero singular values, and all transformations requiring additional independent directions cannot be represented by the parameterization. Increasing r enlarges the available adaptation space, but simultaneously increases the number of trainable parameters. This creates an inherent trade-off between the expressive capacity of the update and the parameter budget.

This restriction can become important when the adaptation required by a task contains multiple distinct components that cannot be adequately represented within a low-dimensional subspace. In such cases, different adaptation signals must be encoded using the same limited set of singular directions. Although the extent to which this limitation affects a particular task depends on the task and model, the rank ceiling itself is an intrinsic property of the parameterization and cannot be removed without increasing the allowed rank.

Limited spectral energy dispersion. The rank constraint also places an upper bound on the stable rank:

$$\mathrm{Srank}(\Delta W) \le \mathrm{rank}(\Delta W) \le r.$$

Therefore, a low-rank parameterization inherently limits the maximum number of directions that can carry substantial spectral energy. More importantly, the nominal rank alone does not determine how effectively these directions are utilized. Even when r is moderately large, the learned singular values may remain highly concentrated in a small subset of directions, resulting in a stable rank substantially below the prescribed rank.

This distinction is important for interpreting high-rank PEFT methods. Increasing the nominal rank expands the set of representable directions, but does not by itself guarantee that the optimization process will distribute spectral energy broadly across them. The stable rank therefore provides an empirical measure of whether the additional representational capacity is actually utilized by the learned update.

Limited effective spectral utilization. Similarly, the effective rank of a rank-r update cannot exceed r. However, its actual value additionally depends on the distribution of the singular values within the available rank. If the spectral mass is concentrated in a small number of dominant singular values, the normalized distribution p_i becomes highly non-uniform, resulting in a substantially lower effective rank than the nominal rank.

This observation highlights a limitation that is not captured by numerical rank alone. Two methods may produce updates with similar numerical ranks while exhibiting substantially different effective ranks, indicating different degrees of spectral concentration. Consequently, evaluating only the number of non-zero or above-threshold singular values may overestimate the effective utilization of the adaptation space. Stable rank and effective rank provide complementary evidence for determining whether a high-rank parameterization genuinely exploits a broad range of spectral directions.

Summary Explicit low-rank parameterization imposes a hard ceiling on the dimensionality of the adaptation space and consequently constrains the maximum achievable numerical, stable, and effective ranks. Moreover, a high nominal rank does not necessarily imply broad spectral utilization, since the singular-value distribution may remain highly concentrated. These observations motivate a different design principle for PEFT: rather than obtaining parameter efficiency by explicitly restricting the rank of the update, parameter efficiency can instead be achieved through structured parameter sharing while leaving the rank of the resulting update unconstrained. This principle motivates the design of HRHA.

C Theory Supplementary

C.1 Rank Properties

Theorem 5 (Rank Freedom of HRHA)**.** *Let $\Delta W \in \mathbb{R}^{m\times n}$ be parameterized by HRHA with block size b, shared orthogonal matrices U and V, and diagonal block coefficients Σ_{ij}. Then HRHA does not impose a prescribed low-rank upper bound on $\operatorname{rank}(\Delta W)$ beyond the intrinsic dimensional bound*

$$\operatorname{rank}(\Delta W) \le \min(m, n). \tag{7}$$

Moreover, the maximum rank $\min(m, n)$ is attainable by a suitable choice of the block coefficients. Hence, HRHA is rank-unconstrained in the sense that it can represent a full-rank update whenever the matrix dimensions permit it.

Proof. From Equation equation 3 we have the factorization

$$\Delta W = (I_{m'} \otimes U)\, \tilde{\Sigma}\, (I_{n'} \otimes V)^\top.$$

The matrices U and V are orthogonal and therefore invertible. Because the Kronecker product of invertible matrices is invertible, both $I_{m'} \otimes U$ and $I_{n'} \otimes V$ are invertible. A standard property of matrix rank states that for any invertible matrices L and R,

$$\operatorname{rank}(LXR) = \operatorname{rank}(X).$$

Applying this to the factorization yields

$$\operatorname{rank}(\Delta W) = \operatorname{rank}\big((I_{m'} \otimes U)\, \tilde{\Sigma}\, (I_{n'} \otimes V)^\top\big) = \operatorname{rank}(\tilde{\Sigma}). \tag{8}$$

Thus the rank of ΔW is entirely determined by the rank of $\tilde{\Sigma}$.

We now analyse the structure of $\tilde{\Sigma}$. Each diagonal block Σ_{ij} can be written as

$$\Sigma_{ij} = \operatorname{diag}\big(\sigma_{ij}^{(1)}, \sigma_{ij}^{(2)}, \ldots, \sigma_{ij}^{(b)}\big). \tag{9}$$

Consequently, the entries of $\tilde{\Sigma}$ are partitioned into b independent groups, each indexed by $k = 1, \ldots, b$, consisting of the values $\{\sigma_{ij}^{(k)}\}$. By permuting rows and columns so that all entries with the same k are grouped together, there exist permutation matrices P and Q such that

$$P\tilde{\Sigma}Q = \operatorname{diag}\big(A^{(1)}, A^{(2)}, \ldots, A^{(b)}\big), \tag{10}$$

where each $A^{(k)} \in \mathbb{R}^{m' \times n'}$ is given by

$$A^{(k)} = \begin{bmatrix} \sigma_{11}^{(k)} & \cdots & \sigma_{1n'}^{(k)} \\ \vdots & \ddots & \vdots \\ \sigma_{m'1}^{(k)} & \cdots & \sigma_{m'n'}^{(k)} \end{bmatrix}. \tag{11}$$

Permutation matrices are invertible, so they preserve rank:

$$\operatorname{rank}(\tilde{\Sigma}) = \operatorname{rank}(P\tilde{\Sigma}Q). \tag{12}$$

The rank of a block-diagonal matrix equals the sum of the ranks of its diagonal blocks. Hence

$$\operatorname{rank}(\tilde{\Sigma}) = \sum_{k=1}^{b} \operatorname{rank}\big(A^{(k)}\big). \tag{13}$$

Since $A^{(k)}$ has dimensions $m' \times n'$, its rank cannot exceed $\min(m', n')$:

$$\operatorname{rank}(A^{(k)}) \leq \min(m', n').$$

Substituting this into Equation equation 13 gives the upper bound

$$\operatorname{rank}(\tilde{\Sigma}) \leq b \, \min(m', n').$$

Using $m = m'b$ and $n = n'b$, we obtain $b \, \min(m', n') = \min(m, n)$, and therefore

$$\operatorname{rank}(\tilde{\Sigma}) \leq \min(m, n).$$

Combined with Equation equation 8, this implies

$$\operatorname{rank}(\Delta W) \leq \min(m, n). \tag{14}$$

The bound equation 14 is exactly the maximum possible rank of any $m \times n$ matrix. Moreover, by choosing each $A^{(k)}$ to have full rank $\min(m', n')$, we can make $\operatorname{rank}(\tilde{\Sigma})$ equal to $\min(m, n)$, and consequently $\operatorname{rank}(\Delta W) = \min(m, n)$. Thus, the HRHA parameterization can realise any rank between 1 and $\min(m, n)$, imposing no rank restriction whatsoever. □

C.2 Rank Verification

We evaluate the spectral characteristics of the weight updates produced by HRHA, LoRA, LoHa, HiRA, GraLoRA, PoLAR, SVFT, and full fine-tuning (FFT) on the Qwen2.5-0.5B model using the three complementary rank metrics introduced in Section B.1: numerical rank, stable rank, and effective rank. The analysis is conducted on five representative weight matrices: query, key, value, feed-forward up-projection (Up), and feed-forward down-projection (Down). The summary results are visualized in Fig. 1 and the detailed figures of each method are provided in Fig. 3.

Numerical rank. The numerical rank directly reflects the number of singular directions whose magnitudes exceed the threshold $\varepsilon = 5 \times 10^{-3}$. As expected, LoRA is strictly confined to the preset rank budget, with numerical ranks of 32 for Q, Up, and Down, and only 13–28 for K and V. LoHa lifts this ceiling to some extent, yet its active subspaces remain heavily constrained: numerical ranks reach only 84–163 for Q, 13–44 for K, 23–53 for V, and 219–256 for Up/Down. GraLoRA and PoLAR are similarly bounded, with numerical ranks mostly below 64 and around 30, respectively, across all modules. HiRA also exhibits a markedly low numerical rank (2–33 for K, 0–33 for V, 11–138 for Q), despite its competitive effective rank reported below. In contrast, HRHA activates a substantially larger number of dimensions. For the large Up and Down matrices, it consistently attains the maximum possible numerical rank of 896, matching the capacity of FFT. On the smaller K and V modules, HRHA covers the vast majority of available directions (93–122 for K, 106–127 for V), whereas FFT occasionally reaches the full 128 but exhibits larger variance. For the Q module, HRHA's numerical rank of 457–545 is somewhat lower than that of FFT (485–754) and SVFT (763–775); nevertheless, it still exceeds the ranks of LoHa, GraLoRA, PoLAR, and LoRA by a factor of 3–10, indicating that no severe dimensionality bottleneck is imposed.

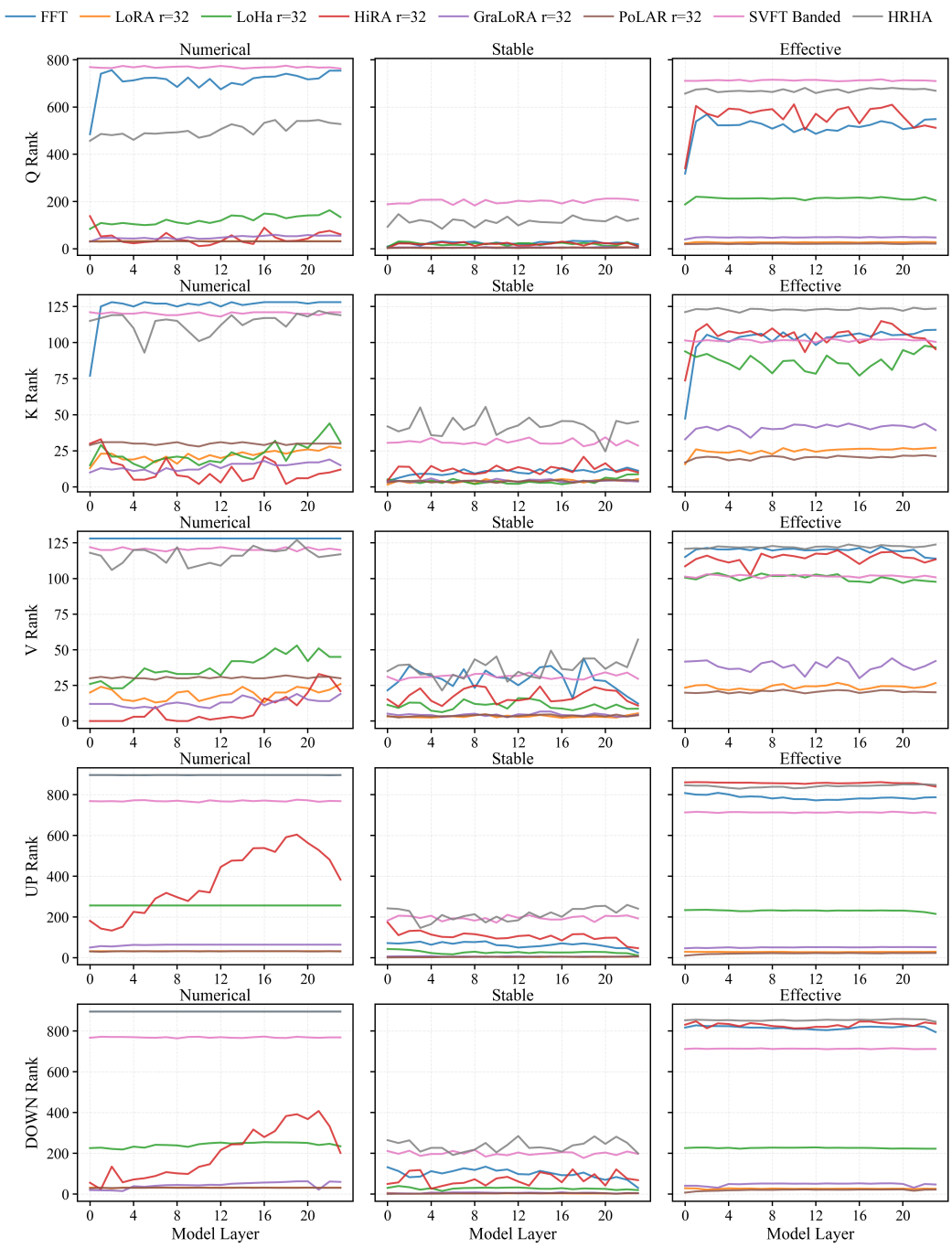

Figure 3: The detailed rank analysis of each methods.

Stable rank. The stable rank measures the spread of spectral energy relative to the dominant singular value. Across the K, V, Up, and Down modules, HRHA achieves the highest stable ranks by a substantial margin. In the Q module, its stable rank ranges from 84 to 146, compared to only 9–33 for FFT, 6–31 for LoHa, 3–30 for HiRA, and 2–7 for LoRA, while SVFT reaches a higher range of 183–212. In the Up and Down modules, the advantage is even more pronounced: HRHA attains 146–285, whereas FFT reaches only 24–135, HiRA 27–174, LoHa 10–43, and LoRA remains below 7. Such high stable ranks signify that the singular values decay very slowly, preventing the update from collapsing into a few dominant directions and thereby preserving a broad set of fine-grained, task-specific transformations. Notably, even FFT, despite accessing the full parameter space, concentrates its energy into a relatively small number of components, resulting in stable ranks that are markedly lower than those of HRHA.

Effective rank. The effective rank captures the uniformity of the normalized singular value distribution. HRHA consistently achieves effective ranks close to the theoretical maximum attainable for its numerical rank, indicating a highly balanced utilization of the active dimensions. In the Q module, its effective rank of 656–680 dramatically exceeds that of FFT (318–568), LoHa (187–220), and LoRA (22–28), and is second only to SVFT (709–717). In the K module, HRHA reaches 120–124, the highest among all methods, compared to 47–108 for FFT, 77–97 for LoHa, 100–103 for SVFT, and below 27 for LoRA; the same holds for V, where HRHA attains 120–124 versus 114–122 for FFT. For the large Up and Down matrices, HRHA's effective rank (829–851 and 846–859, respectively) again dominates all methods except HiRA, which reaches a comparable range on Up (840–861) but with only a fraction of the numerical support. This uniform spectral profile implies that the adapter does not over-specialize on a few principal components, but instead distributes its representational capacity evenly across all available singular directions, which is conducive to better generalization and robustness.

Summary. The simultaneous achievement of high numerical rank, high stable rank, and high effective rank is a distinctive property of HRHA. While FFT activates many dimensions, its energy remains concentrated, leading to moderate stable and effective ranks. Low-rank adapters, by design, underperform on all three metrics; HiRA achieves a superficially high effective rank only through a long tail of negligible singular values, as evidenced by its very low numerical rank. SVFT is the closest competitor, surpassing HRHA on the Q module, but trailing on the K, V, Up, and Down matrices. In contrast, HRHA produces weight updates that are at once high-dimensional, well-dispersed, and maximally uniform. This spectral signature suggests that HRHA can capture the full complexity of downstream tasks within a parameter-efficient structure, and it may even induce a more regularized and generalizable update than full fine-tuning.

C.3 Convexity

Theorem 6 (Convexity). *Consider a linear layer with the squared loss* $\mathcal{L}(\tilde{\Sigma}) = \frac{1}{2}\left\|(W_0 + L\tilde{\Sigma}R)X - Y\right\|_F^2$. *Then* $\mathcal{L}(\tilde{\Sigma})$ *is a convex quadratic function of* $\tilde{\Sigma}$. *More precisely, letting* $\tilde{\sigma} = \mathrm{vec}(\tilde{\Sigma})$, *the objective can be written as* $\mathcal{L}(\tilde{\sigma}) = \frac{1}{2}\|A\tilde{\sigma} - y\|_2^2$, *where* $A = X^\top R^\top \otimes L, y = \mathrm{vec}(Y - W_0X)$. *Consequently,* $\nabla_{\tilde{\sigma}}^2\mathcal{L} = A^\top A \succeq 0$, *and hence* $\mathcal{L}$ *is convex.*

Proof. We first isolate the part of the output that needs to be fitted by the HRHA. Define the residual target

$$E = Y - W_0X \in \mathbb{R}^{m\times d}. \tag{15}$$

Then the loss becomes

$$\mathcal{L}(\tilde{\Sigma}) = \frac{1}{2}\left\|L\tilde{\Sigma}RX - E\right\|_F^2. \tag{16}$$

Following the Theorem 2, the mapping

$$\mathcal{T} : \tilde{\Sigma} \mapsto L\tilde{\Sigma}RX \tag{17}$$

is a linear mapping from $\mathbb{R}^{m\times n}$ to $\mathbb{R}^{m\times d}$. Indeed, for arbitrary $\tilde{\Sigma}_1, \tilde{\Sigma}_2 \in \mathbb{R}^{m\times n}$ and scalars a, b, we have

$$\begin{aligned}\mathcal{T}(a\tilde{\Sigma}_1 + b\tilde{\Sigma}_2) &= L(a\tilde{\Sigma}_1 + b\tilde{\Sigma}_2)RX \\ &= aL\tilde{\Sigma}_1 RX + bL\tilde{\Sigma}_2 RX \\ &= a\mathcal{T}(\tilde{\Sigma}_1) + b\mathcal{T}(\tilde{\Sigma}_2).\end{aligned} \tag{18}$$

Thus, Eq. equation 16 is the squared Euclidean norm of an affine function of $\tilde{\Sigma}$.

To make this structure explicit, we vectorize the matrices. Using the standard identity

$$\operatorname{vec}(ABC) = (C^\top \otimes A)\operatorname{vec}(B), \tag{19}$$

we obtain

$$\operatorname{vec}(L\tilde{\Sigma}RX) = (X^\top R^\top \otimes L)\operatorname{vec}(\tilde{\Sigma}). \tag{20}$$

Let

$$\tilde{\sigma} = \operatorname{vec}(\tilde{\Sigma}) \in \mathbb{R}^{mn}, \qquad A = X^\top R^\top \otimes L \in \mathbb{R}^{md\times mn}, \tag{21}$$

and

$$y = \operatorname{vec}(E) = \operatorname{vec}(Y - W_0 X) \in \mathbb{R}^{md}. \tag{22}$$

Because the Frobenius norm is equivalent to the Euclidean norm after vectorization,

$$\|M\|_F = \|\operatorname{vec}(M)\|_2,$$

Eq. equation 16 can therefore be rewritten as

$$\mathcal{L}(\tilde{\sigma}) = \frac{1}{2}\|A\tilde{\sigma} - y\|_2^2. \tag{23}$$

We next show explicitly that this objective is quadratic. Expanding Eq. equation 23 gives

$$\begin{aligned}\mathcal{L}(\tilde{\sigma}) &= \frac{1}{2}(A\tilde{\sigma} - y)^\top(A\tilde{\sigma} - y) \\ &= \frac{1}{2}\tilde{\sigma}^\top A^\top A\tilde{\sigma} - y^\top A\tilde{\sigma} + \frac{1}{2}y^\top y.\end{aligned} \tag{24}$$

Thus, $\mathcal{L}$ is a quadratic function of the trainable parameter $\tilde{\sigma}$.

Taking the gradient with respect to $\tilde{\sigma}$ yields

$$\nabla_{\tilde{\sigma}}\mathcal{L} = A^\top(A\tilde{\sigma} - y), \tag{25}$$

and taking the derivative once more gives the Hessian

$$\nabla^2_{\tilde{\sigma}}\mathcal{L} = A^\top A. \tag{26}$$

For any vector $z \in \mathbb{R}^{mn}$,

$$z^\top \nabla^2_{\tilde{\sigma}}\mathcal{L} z = z^\top A^\top A z = \|Az\|_2^2 \geq 0. \tag{27}$$

Therefore,

$$\nabla^2_{\tilde{\sigma}}\mathcal{L} = A^\top A \succeq 0. \tag{28}$$

By the second-order characterization of convex functions, $\mathcal{L}(\tilde{\sigma})$ is convex. □

C.4 Global Convergence

Theorem 7 (Global Convergence). *Following the setting of Theorem 3, let $\mathcal{L}(\tilde{\sigma}) = \frac{1}{2}\|A\tilde{\sigma} - y\|_2^2$, and denote $H = A^\top A$. Let $\tilde{\sigma}^{(t)}$ be generated by gradient descent: $\tilde{\sigma}^{(t+1)} = \tilde{\sigma}^{(t)} - \eta\nabla\mathcal{L}(\tilde{\sigma}^{(t)})$, where $0 < \eta < \frac{2}{\lambda_{\max}(H)}$. Then: 1) $\mathcal{L}(\tilde{\sigma}^{(t)})$ converges monotonically to the global minimum $\mathcal{L}^\star$. 2) The iterates $\tilde{\sigma}^{(t)}$ converge to a global minimizer of $\mathcal{L}$. 3) If A has full column rank, equivalently $H \succ 0$, the global minimizer is unique, and the convergence is linear: $\|\tilde{\sigma}^{(t)} - \tilde{\sigma}^\star\|_2 \leq \rho^t\|\tilde{\sigma}^{(0)} - \tilde{\sigma}^\star\|_2$, where $\rho = \max_i |1 - \eta\lambda_i(H)| < 1$.*

Proof. Since

$$\nabla \mathcal{L}(\tilde{\sigma}) = A^\top (A\tilde{\sigma} - y) = H\tilde{\sigma} - A^\top y, \tag{29}$$

the gradient descent update becomes

$$\tilde{\sigma}^{(t+1)} = (I - \eta H)\tilde{\sigma}^{(t)} + \eta A^\top y. \tag{30}$$

Let $\tilde{\sigma}^\star$ be any global minimizer. Because the objective is convex and differentiable, it satisfies the first-order optimality condition

$$H\tilde{\sigma}^\star = A^\top y. \tag{31}$$

Subtracting this equation from Eq. 30 gives

$$\tilde{\sigma}^{(t+1)} - \tilde{\sigma}^\star = (I - \eta H)(\tilde{\sigma}^{(t)} - \tilde{\sigma}^\star). \tag{32}$$

Since $H = A^\top A \succeq 0$, all eigenvalues of H are nonnegative. Under the step-size condition

$$0 < \eta < \frac{2}{\lambda_{\max}(H)},$$

all nonzero eigenvalues satisfy

$$|1 - \eta\lambda_i(H)| < 1.$$

Thus, the error components corresponding to the positive eigenvalues decay geometrically.

For the components in the null space of H, we have

$$Hz = 0 \quad \Longrightarrow \quad Az = 0.$$

Moreover,

$$z^\top A^\top y = (Az)^\top y = 0.$$

Hence, the gradient has no component in $\ker(H)$, and gradient descent leaves the null-space component of the initialization unchanged. Consequently, the iterates converge to a global minimizer, with the particular minimizer determined by the initialization.

If A has full column rank, then $H \succ 0$ and

$$0 < \lambda_{\min}(H) \le \lambda_i(H) \le \lambda_{\max}(H).$$

Therefore,

$$\rho = \|I - \eta H\|_2 = \max_i |1 - \eta\lambda_i(H)| < 1.$$

It follows that

$$\|\tilde{\sigma}^{(t)} - \tilde{\sigma}^\star\|_2 \le \rho^t \|\tilde{\sigma}^{(0)} - \tilde{\sigma}^\star\|_2, \tag{33}$$

which establishes linear convergence to the unique global minimizer. □

C.5 Initialization of Left and Right Bases

To ensure the representational capability of HRHA, the shared matrices U and V are required to be orthogonal, as discussed in the theoretical analysis. Orthogonality guarantees invertibility and preserves the rank properties of the parameterization. We consider two orthogonal initialization strategies for U and V: random orthogonal initialization and joint bi-sided principal component analysis (PCA). Ablation results show that both strategies yield competitive performance, whereas non-orthogonal initialization leads to a substantial degradation in performance, highlighting the importance of maintaining orthogonality.

Random Orthogonal Initialization. We adopt the orthogonal initialization method described inSaxe et al. (2013). Specifically, we first sample a random matrix $M \in \mathbb{R}^{b\times b}$ whose entries are drawn independently from a standard normal distribution. We then perform QR decomposition:

$$M = QR,$$

where Q is orthogonal and R is upper triangular. To ensure a uniform distribution, we multiply Q element-wise by a matrix constructed from the signs of the diagonal entries of R, following the procedure described inMezzadri (2006). The resulting matrix Q is an orthogonal matrix with a symmetric distribution. Both U and V are initialized independently using this procedure.

Joint Bi-sided PCA. The initialization of the shared bases U and V is critical to the expressive capability of HRHA. Since U and V define the global left and right subspaces in which all block-wise adaptations are expressed, a principled initialization is essential for capturing the dominant structural patterns of the pre-trained weights. To this end, we design a joint bi-sided principal component analysis (PCA) procedure over all blocks extracted from the pre-trained parameter matrix.

Let $\{A_i\}_{i=1}^N$, where $A_i \in \mathbb{R}^{b\times b}$, denote the collection of blocks obtained by partitioning the pre-trained weight matrix W_0. Our objective is to determine shared orthogonal matrices $U \in \mathbb{R}^{b\times b}$ and $V \in \mathbb{R}^{b\times b}$ such that each block can be approximated as

$$A_i \approx U\Sigma_i V^\top,$$

where Σ_i is a block-specific coefficient matrix. This formulation seeks common left and right subspaces that jointly explain the variations across all blocks.

To estimate these shared subspaces, we construct the following second-order statistical matrices:

$$C_r = \sum_{i=1}^{N} A_i A_i^\top, \qquad C_c = \sum_{i=1}^{N} A_i^\top A_i.$$

Here, C_r represents the aggregated row-space covariance matrix, while C_c represents the aggregated column-space covariance matrix across all blocks. These matrices characterize the global energy distribution along row and column directions, respectively.

Specifically, each term $A_i A_i^\top = \sum_j a_{ij} a_{ij}^\top$, where a_{ij} denotes the j-th column of A_i, corresponds to accumulating the outer products of column vectors. Thus, C_r is equivalent to concatenating all column vectors from all blocks and performing PCA on the resulting dataset. Similarly, C_c aggregates the outer products of row vectors, corresponding to PCA over all row directions.

We then perform eigenvalue decomposition:

$$C_r = U\Lambda_r U^\top, \qquad C_c = V\Lambda_c V^\top,$$

where Λ_r and Λ_c are diagonal matrices containing the eigenvalues. Because C_r and C_c are symmetric positive semi-definite by construction, their eigenvectors form orthogonal matrices. The resulting matrices U and V therefore capture the principal row and column subspaces of the pre-trained blocks, providing an informed and data-driven initialization for the HRHA.

C.6 Parameter Efficiency

In this section, we analyze the number of trainable parameters introduced by HRHA. For a weight update matrix $\Delta W \in \mathbb{R}^{m\times n}$, HRHA optimizes only the block coefficient matrix

$$\tilde{\Sigma} = \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1n'} \\ \vdots & \ddots & \vdots \\ \Sigma_{m'1} & \cdots & \Sigma_{m'n'} \end{bmatrix} \in \mathbb{R}^{m\times n}, \tag{34}$$

where each $\Sigma_{ij} \in \mathbb{R}^{b\times b}$ is diagonal and $m' = m/b$, $n' = n/b$. Since each diagonal block contains b trainable parameters and there are $m'n' = \frac{mn}{b^2}$ blocks, the total number of trainable parameters is mn/b.

Comparison with Fully Fine-Tuning (FT). In fully fine-tuning, all entries of ΔW are updated, resulting in mn trainable parameters. In contrast, HRHA reduces the number of trainable parameters by a factor of b. In practice, b is typically set to 128 or 256, leading to a substantial reduction in parameter count compared to fully fine-tuning.

Comparison with LoRA. For LoRA, the weight update $\Delta W \in \mathbb{R}^{m\times n}$ is parameterized as $\Delta W = BA$, where $B \in \mathbb{R}^{m\times r}$ and $A \in \mathbb{R}^{r\times n}$. The total number of trainable parameters is therefore $(m+n)r$.

HRHA uses fewer parameters than LoRA whenever $\frac{mn}{b} < (m+n)r$, which is equivalent to

$$b > \frac{mn}{(m+n)r}.$$

Empirically, we observe that under such configurations, HRHA can achieve performance comparable to or exceeding that of LoRA while using fewer trainable parameters.

C.7 Computational Overhead

Theoretical time complexity. The Kronecker products in Equation equation 3 have a block-diagonal structure: $I_{m'} \otimes U = \mathrm{diag}(U, \ldots, U)$ and $(I_{n'} \otimes V)^\top = \mathrm{diag}(V^\top, \ldots, V^\top)^\top$. This structure enables efficient block-wise assembly of ΔW. Left multiplication by $I_{m'} \otimes U$ applies U to each block row of $\tilde{\Sigma}$, while right multiplication by $(I_{n'} \otimes V)^\top$ applies $V^\top$ to each block column. The computation thus decomposes into $m' \times n'$ independent blocks, each of the form $U\Sigma_{ij}V^\top$, where $\Sigma_{ij} \in \mathbb{R}^{b\times b}$ is diagonal.

For a single block, scaling the columns of U by Σ_{ij} costs $O(b^2)$ operations, and the subsequent multiplication with $V^\top$ is a standard $b \times b$ matrix product requiring $O(b^3)$ operations. Hence each block incurs $O(b^3)$ operations, and the total complexity for constructing ΔW is

$$O\big(m'n'\,b^3\big) = O\Big(\frac{mn}{b^2}\,b^3\Big) = O(m\,n\,b).$$

By contrast, ignoring the Kronecker structure and computing two dense matrix multiplications directly on the $m \times n$ matrix $\tilde{\Sigma}$ would cost $O(m^2n + mn^2)$. The Kronecker factorisation therefore yields a substantial reduction in computational cost, particularly when the shared dimension b is small relative to m and n.

Implementation and practical efficiency. In practice, the construction of ΔW can be realised with a single tensor contraction:

$$M_{i,j,a,b} = \sum_k U_{a,k}\, C_{i,j,k}\, V_{k,b},$$

followed by a permutation and reshape to form the full matrix,

$$\Delta W = \texttt{M.permute(0,2,1,3).reshape}(m,\, n).$$

Here $C \in \mathbb{R}^{m'\times n'\times b}$ collects the diagonal entries of the blocks Σ_{ij}. For each block (i, j) this computes $U\ \mathrm{diag}(C_{ij})\,V^\top$ with b^3 multiply–add operations, so the total complexity remains $O(mnb)$. Expressing all block operations as a single tensor contraction avoids Python-level loops and fully leverages GPU parallelism and highly optimised linear algebra kernels, resulting in substantially lower wall-clock time than an iterative block-by-block approach.

Since U and V are fixed shared matrices, the constant factor can be further reduced by precomputing the tensor $T_{k,a,b} = U_{a,k}\,V_{k,b}$ and rewriting the contraction as

$$M_{i,j,a,b} = \sum_k C_{i,j,k}\, T_{k,a,b},$$

which replaces the two multiplications involving U and V with a single multiply–add from T, without affecting the asymptotic complexity. With these optimisations, the computational efficiency of HRHA is substantially enhanced. We compared its training time with that of LoRA, and the experimental results show that HRHA achieves training speed comparable to LoRA.

D Experiment Supplementary

D.1 Experimental Settings

Infrastructure. All experiments are conducted on a server equipped with two NVIDIA GeForce RTX 4090 GPUs. The software environment, including the Python version and package dependencies, is specified in the `env.yaml` file provided with the source code.

Hyperparameters. For reproducibility, the detailed hyperparameter configurations for each task are provided in the corresponding experimental sections. To ensure a fair comparison with the baselines, we keep the hyperparameter settings consistent with those of the corresponding baseline methods whenever applicable. We specifically report the block size and learning rate used by HRHA, while other training hyperparameters follow the settings adopted by the respective baselines unless otherwise specified.

Table 8: The number of trainable parameters of each method on commonsense reasoning tasks.

Methods	Qwen2.5-0.5B	Qwen2.5-3B	Phi3-3B	Llama3.2-3B	Qwen2.5-7B
LoRA16	5.9M	20.1M	22.0M	16.5M	27.1M
LoRA32	11.8M	40.1M	44.0M	33.0M	54.1M
LoHa16	5.9M	20.1M	22.0M	16.5M	27.1M
LoHa32	11.8M	40.1M	44.0M	33.0M	54.1M
HiRA16	5.9M	20.1M	22.0M	16.5M	27.1M
HiRA32	11.8M	40.1M	44.0M	33.0M	54.1M
GraLoRA16	5.9M	20.1M	22.0M	16.5M	27.1M
GraLoRA32	11.8M	40.1M	44.0M	33.0M	54.1M
PoLAR16	5.9M	20.1M	22.0M	16.5M	27.1M
PoLAR32	11.8M	40.1M	44.0M	33.0M	54.1M
SVFT	1.2M	4.1M	5.0M	5.4M	/
HRHA	**1.83M**	**14.2M**	**13.0M**	**14.5M**	**16.7M**

Baselines. We compare HRHA with a diverse set of representative PEFT methods, including LoRA Hu et al. (2022), LoHa Hyeon-Woo et al. (2021), HiRA Huang et al. (2025), GraLoRA Jung et al. (2026), PoLAR Lion et al. (2026), and SVFT Lingam et al. (2024). These baselines cover conventional low-rank adaptation, high-rank parameterizations, block-wise adaptation, manifold-based adaptation, and spectral parameterization. A brief description of their key ideas is provided in the Related Work section (Sec. A).

Source Code. The source code, environment configuration, and scripts required to reproduce our experiments are provided in the supplementary materials.

D.2 Commonsense Reasoning.

We evaluate HRHA on eight widely used commonsense reasoning benchmarks: BoolQ Clark et al. (2019) (yes/no question answering), PIQA Bisk et al. (2020) (physical commonsense reasoning), SIQA Sap et al. (2019) (social commonsense reasoning), HellaSwag Zellers et al. (2019) (multiple-choice scenario completion), WinoGrande Sakaguchi et al. (2021) (binary pronoun resolution), ARC-c and ARC-e Clark et al. (2018) (challenging and easy grade-school science questions), and OBQA Mihaylov et al. (2018) (multi-step reasoning). Collectively, these datasets span a broad spectrum of reasoning skills, including natural language understanding, physical and social inference, and multi-hop reasoning. For additional background on these benchmarks, we refer readers to the survey in Hu et al. (2023). Experiments are conducted on multiple large language models, including Qwen2.5 (0.5B, 3B, and 7B), Phi-3 (3B), and LLaMA-3.2 (3B). This setup enables comparisons across different scales within the same architecture (e.g., Qwen2.5-0.5B, Qwen2.5-3B, and Qwen2.5-7B), as well as across distinct architectures with comparable parameter sizes (e.g., Qwen2.5-3B, Phi-3-3B, and LLaMA-3.2-3B). HRHA is applied to the attention projection matrices (W_q, W_k, W_v) and the feed-forward layers (W_{up}, W_{down}). Following Hu et al. (2022), we assess data efficiency by fine-tuning on both the full training set (170K samples) and a reduced subset (15K samples). We compare HRHA with LoRA and its variants using $r = 16$ and $r = 32$ adapters. All models are trained for one epoch. For HRHA, the block size is set to 128×128, and the learning rate is 10^{-3}. Aggregated results are reported in Table 1, with the corresponding number of trainable parameters shown in Table 8.

Table 1 reports the average accuracy over eight commonsense reasoning benchmarks for five model families under two data regimes. HRHA achieves consistently strong results, particularly in the low-resource setting, where it outperforms all competing methods on every evaluated model.

Under the 15K low-data regime, HRHA attains the highest accuracy across all five models, with improvements of up to 4.58 percentage points over the strongest baseline (on Qwen2.5-0.5B: 56.14 vs. LoRA32's 51.56). This advantage is especially notable on LLaMA-3.2-3B, where HRHA reaches 71.85, while the best LoRA variant achieves only 68.12. In contrast, several higher-rank methods exhibit severe instability or degradation under limited data: HiRA collapses to 9.28 on LLaMA-3.2-3B, and SVFT, despite its parameter efficiency, lags substantially behind on all models. These

observations indicate that HRHA's full-rank block-wise parameterization provides robust adaptation even when training samples are scarce.

When trained on the full 170K dataset, HRHA remains highly competitive. It obtains the best accuracy on Qwen2.5-3B (82.86) and Phi3-3B (84.42), and on the remaining three models it is within 0.52–1.14 points of the best result, which is achieved by LoRA32. Crucially, these small gaps are obtained without the aggressive rank inflation used by LoRA32; HRHA maintains a compact parameterization while still matching or approaching the highest scores. The higher-rank baselines, such as LoHa, HiRA, GraLoRA, and PoLAR, fail to deliver consistent gains: for example, PoLAR32 underperforms on Qwen2.5-3B (80.29) and shows a large drop on LLaMA-3.2-3B (70.95), and GraLoRA32 performs worse than its rank-16 counterpart on several models, suggesting that merely increasing rank does not guarantee improved commonsense reasoning.

Overall, the results demonstrate that HRHA offers a favorable trade-off between data efficiency and task performance. Its superiority in low-resource regimes, combined with competitive full-data results, highlights the benefit of removing low-rank constraints while maintaining structured parameter sharing, and confirms the applicability of HRHA across diverse model architectures and scales.

D.3 Natural Language Understanding

We evaluate HRHA on the General Language Understanding Evaluation (GLUE) benchmark Wang et al. (2018), which comprises a diverse collection of natural language understanding (NLU) tasks. Following prior work Li et al. (2024), we report results on six representative GLUE tasks: CoLA Warstadt et al. (2019), SST-2 Socher et al. (2013), MRPC Dolan & Brockett (2005), STS-B Cer et al. (2017), QNLI Rajpurkar et al. (2018), and RTE. Experiments are conducted using RoBERTa$_{\text{base}}$ and RoBERTa$_{\text{large}}$ Liu et al. (2019). HRHA is applied to all linear transformation modules, including the attention projections (W_q, W_k, W_v, W_o) and the feed-forward layers (W_{up}, W_{down}). All models are trained for 10 epochs, and the learning rate for HRHA is set to 10^{-3} with a block size of 256×256.

Table 2 summarizes the GLUE results for RoBERTa$_{\text{base}}$ and RoBERTa$_{\text{large}}$, with all baseline methods using rank 16. HRHA consistently achieves the highest or near-highest performance while requiring substantially fewer trainable parameters.

On RoBERTa$_{\text{base}}$, HRHA obtains the best average score of 82.6, outperforming LoRA (82.5) and GraLoRA (82.2) by a small but consistent margin, and surpassing PoLAR, SVFT, LoHa, and HiRA by larger gaps. Notably, this result is achieved with only 0.33M parameters, which is about $8\times$ fewer than the rank-16 baselines (2.65M) and even lower than the highly compressed SVFT (0.93M). The advantages are especially clear on MRPC (87.7) and CoLA (59.7), where HRHA achieves the best scores.

On RoBERTa$_{\text{large}}$, HRHA again attains the highest average (86.5) and the best results on five out of six tasks, including an RTE score of 84.0 that exceeds LoRA by 1.8 points and GraLoRA by 2.0 points, despite using only 1.18M parameters compared to 7.07M for the baselines. Although GraLoRA is competitive overall (86.2), it lags behind on RTE. SVFT achieves a moderate average (79.3) but falls far behind on CoLA and RTE, indicating that its aggressive parameter reduction limits task adaptation.

Among the higher-rank baselines, LoHa and PoLAR fail to reach the performance of LoRA or HRHA, especially on RTE and CoLA, while HiRA suffers from complete training instability, with near-zero scores on CoLA and STS-B. These observations indicate that merely increasing the nominal rank does not guarantee robust expressive capacity.

Overall, HRHA delivers the best average performance on both model scales with the fewest trainable parameters, demonstrating that its structured full-rank parameterization effectively balances expressiveness and efficiency, particularly on difficult NLU tasks such as RTE and CoLA.

D.4 Image Classification

We evaluate methods on image classification using the Vision Transformer (ViT) Dosovitskiy et al. (2020), a representative foundation model in computer vision. Experiments are conducted on

Table 9: Hyperparameter settings for ViT-Base and ViT-Large on the image classification datasets. *lr*: learning rate, *head_lr*: learning rate of classifier head, *wd*: weight decay.

	ViT-Base			ViT-Large		
Dataset	lr	head_lr	wd	lr	head_lr	wd
OxfordPets	0.005	0.005	0.0005	0.001	0.0005	0.0005
StanfordCars	0.01	0.005	0.0	0.01	0.005	0.1
CIFAR10	0.01	0.005	0.05	0.01	0.05	0.1
CIFAR100	0.01	0.005	0.05	0.01	0.05	0.05
EuroSAT	0.01	0.0005	0.05	0.01	0.0005	0.01
FGVC	0.01	0.005	0.0005	0.01	0.0005	0.0005
RESISC45	0.01	0.005	0.0005	0.01	0.0005	0.1

seven public benchmark datasets: OxfordPets[1] , StanfordCars[2], CIFAR10[2], EuroSAT[3], GVC[2], RESISC45[4], and CIFAR100[2]. For each baseline, we adopt the hyperparameter settings recommended in the original papers (e.g., rank r). To ensure a fair comparison, we closely follow the experimental protocols described in Hu et al. (2022); Meng et al. (2024); Gao et al. (2024). All models are trained for 10 epochs, and the target modules are the attention projection matrices (W_q and W_v). For HRHA, the block size is set to 128×128, and the learning rate is shown in table 9.

Table 3 reports image classification results on seven datasets for ViT-Base and ViT-Large, with all PEFT baselines using rank 16 and each method trained with its optimal hyperparameters. HRHA achieves the strongest overall parameter efficiency and the highest or near-highest accuracy among all PEFT methods, closing most of the gap to full fine-tuning (FFT) while using orders of magnitude fewer trainable parameters.

On ViT-Base, HRHA attains an average accuracy of 87.66%, which is essentially on par with FFT (87.74%) and surpasses all PEFT baselines by a clear margin: HiRA achieves 84.51%, SVFT 83.27%, LoHa 80.78%, GraLoRA 78.68%, LoRA 78.22%, and PoLAR 75.12%. Most strikingly, this performance is obtained with only 0.26M trainable parameters, which is about $2.8\times$ fewer than LoRA, LoHa, HiRA, and GraLoRA (0.74M), $1.8\times$ fewer than SVFT (0.46M), and approximately $330\times$ fewer than FFT (85.8M).

The advantage of HRHA is most evident on challenging fine-grained datasets. On StanfordCars, it achieves 79.73%, substantially outperforming HiRA (63.17%), LoHa (59.82%), SVFT (58.31%), GraLoRA (41.47%), LoRA (39.72%), and PoLAR (28.75%). On FGVC, HRHA reaches 56.98%, again the best among PEFT methods and ahead of HiRA (51.23%), SVFT (48.62%), and LoHa (35.42%). On CIFAR10, HRHA attains the highest accuracy of 98.88%, and on OxfordPets it ranks second at 93.68%, closely trailing FFT (98.92%) but exceeding all other PEFT methods. These results indicate that HRHA is particularly effective when the task requires rich, spatially localized feature transformations.

On ViT-Large, HRHA achieves an average of 90.23% with only 0.59M parameters, placing it second among all methods behind HiRA (91.54%) and FFT (91.40%), and ahead of SVFT (90.80%), GraLoRA (90.13%), LoRA (89.13%), PoLAR (86.34%), and LoHa (85.06%). Although HiRA attains a slightly higher average, it requires 1.77M parameters, which is three times more than HRHA. On RESISC45, HRHA reaches 95.08%, closely matching HiRA (95.87%) and GraLoRA (95.64%), and on CIFAR100 it achieves 93.68%, second only to HiRA (93.71%) and FFT (88.90%). On OxfordPets, HRHA attains 94.65%, the highest among all methods including FFT, and on EuroSAT it reaches 98.97%, matching FFT's 96.43% by a clear margin. The only relatively weaker results are on StanfordCars and FGVC, where HiRA and SVFT slightly outperform HRHA, though the gaps are small (83.95% vs. 88.29% on Cars and 66.25% vs. 71.34% on FGVC).

[1]https://huggingface.co/datasets/timm/oxford-iiit-pet
[2]https://huggingface.co/datasets/Multimodal-Fatima
[3]https://huggingface.co/datasets/timm/eurosat-rgb
[4]https://huggingface.co/datasets/timm/resisc45

The parameter efficiency of HRHA is a decisive advantage. On ViT-Base, it uses only 0.26M parameters, less than half of SVFT and roughly one-third of the rank-16 baselines. On ViT-Large, it requires only 0.59M parameters, compared to 1.03M for SVFT and 1.77M for LoRA, LoHa, HiRA, and GraLoRA. Despite this compactness, HRHA consistently ranks among the top two PEFT methods on both backbones.

Among the higher-rank baselines, HiRA and SVFT are the closest competitors, but their advantages are inconsistent across datasets and model scales. HiRA performs well on ViT-Large but is clearly behind HRHA on ViT-Base, particularly on StanfordCars and FGVC. SVFT achieves competitive averages but requires approximately twice the parameters of HRHA and underperforms on hard datasets. LoHa, GraLoRA, and PoLAR fail to match the performance of HRHA on any benchmark, and their results degrade sharply on StanfordCars and FGVC.

Overall, the image classification results demonstrate that HRHA offers an exceptional balance between accuracy and parameter efficiency, consistently matching or surpassing state-of-the-art PEFT methods while using the fewest trainable parameters and approaching the performance of full fine-tuning.

D.5 Personalized Dialogue Generation

We evaluated the ability of fine-tuned models to engage in natural conversations using the ConvAI2 dataset Dinan et al. (2019) with 17, 878 training and 1, 000 testing multi-turn conversations. Following the experimental setup of Gu et al. (2026), where only the speaker's persona is revealed (self-persona setting), we report the quality of the generated responses using standard metrics, including BLEU, BERTScore Zhang et al. (2019), METEOR Banerjee & Lavie (2005), and ROUGE Lin (2004). HRHA is applied to the attention projections W_q and W_v. All models are trained for 1 epoch, and the learning rate for HRHA is set to $1e-4$ with a block size of 256×256. The rank of LoRA and its variants is set to 16.

Table 4 reports the results on personalized dialogue generation for Llama-2-7B and Llama-3-8B. HRHA achieves competitive or superior performance on both models while using substantially fewer trainable parameters than most baselines.

On Llama-2-7B, HRHA attains the highest average score of 47.92, slightly surpassing LoHa (47.90) and GraLoRA (47.87), and outperforming LoRA (47.82), SVFT (47.84), and HiRA (47.63) by a small but consistent margin. The advantage is particularly evident in the generation-oriented metrics: HRHA achieves a METEOR score of 13.23, which is the highest among all methods, and an R-L score of 12.93, also the best, while maintaining a BLEU score of 2.39 that is comparable to LoRA (2.57) and LoHa (2.59). The BERT-based similarity metrics (BERT F1, BERT-R, BERT-P) are also among the best, with BERT F1 reaching 86.32, equaling LoHa and exceeding all other methods. This indicates that the dialogue responses generated by HRHA are not only semantically aligned with the references but also lexically diverse and fluent.

On Llama-3-8B, HRHA obtains an average score of 47.39, ranking third behind HiRA (47.54) and LoRA (47.27), and ahead of LoHa (47.25), SVFT (47.18), GraLoRA (47.08), and PoLAR (38.29). The small average gap to HiRA is primarily due to HiRA's higher BERT-P and R-L scores, which emphasize surface-level overlap with the reference. However, HRHA achieves the best BLEU score of 3.47 among all methods, surpassing LoRA (3.40), HiRA (3.26), and GraLoRA (3.31) by clear margins. Similarly, it attains the best METEOR score of 8.90, ahead of LoRA (8.49), HiRA (8.62), and GraLoRA (8.45). These results show that HRHA generates more diverse and creative responses while preserving strong semantic adequacy, which is particularly important for personalized dialogue generation.

The parameter efficiency of HRHA is particularly striking. On Llama-2-7B, it uses only 4.19M trainable parameters, which is about half of LoRA, LoHa, HiRA, and GraLoRA (8.38M), and even less than SVFT (4.45M). On Llama-3-8B, HRHA requires only 2.62M parameters, which is less than SVFT (2.78M) and about $2.6\times$ fewer than LoRA, LoHa, HiRA, and GraLoRA (6.82M). Despite this substantial reduction in trainable capacity, HRHA achieves the highest generation quality on both models and remains highly competitive on overall average, demonstrating that its structured full-rank parameterization is highly parameter-efficient.

Among the higher-rank variants, PoLAR fails catastrophically on both models, with BLEU, METEOR, and R-L near zero and an average below 39, indicating severe training instability. HiRA performs competitively on Llama-3-8B but drops below HRHA on Llama-2-7B, and its advantage is not consistent across metrics or model scales. GraLoRA is broadly comparable to LoRA but does not surpass HRHA on any generation metric. In contrast, HRHA exhibits robust and stable performance across both model families, with particular strength in surface-level generation quality (BLEU and METEOR), which is crucial for dialogue systems.

Overall, these results confirm that HRHA offers an excellent balance between expressive capacity and parameter efficiency for personalized dialogue generation, consistently producing high-quality responses while using significantly fewer trainable parameters than existing PEFT methods.

D.6 Arithmetic Reasoning

We evaluated the arithmetic reasoning capabilities of the fine-tuned model through the Math10k dataset from Cobbe et al. (2021); Koncel-Kedziorski et al. (2016); Ling et al. (2017), consisting of 10, 000 mathematical reasoning examples. We considered six sub-tasks:MultiArith Roy & Roth (2015), GSM8K Cobbe et al. (2021), AddSub Hosseini et al. (2014), AQuA Ling et al. (2017), SingleEq Koncel-Kedziorski et al. (2015), and SVAMP Patel et al. (2021). HRHA is applied to the attention projections W_q and W_v. All models are trained for 3 epochs, and the learning rate for HRHA is set to $2e-3$ with a block size of 128×128. The rank of LoRA and its variants is set to 16.

Table 5 reports the arithmetic reasoning results on Llama-2-7B and Llama-3-8B across six datasets. HRHA achieves consistently strong performance on both model scales while requiring substantially fewer trainable parameters than the rank-based baselines.

On Llama-2-7B, HRHA attains an average accuracy of 63.82, which is statistically on par with the best result of 63.88 obtained by GraLoRA, and clearly exceeds LoRA (63.02), PoLAR (59.24), LoHa (56.13), SVFT (54.60), and HiRA (47.67). Notably, HRHA achieves the highest accuracy on MultiArith (92.67) and AddSub (89.87), demonstrating its strength in multi-step and symbolic arithmetic. On gsm8k, HRHA reaches 34.57, which is competitive with GraLoRA (35.63) and LoRA (36.16), while on AQuA it attains 23.23, close to GraLoRA's best of 23.62. The only minor lag is on SingleEq, where HRHA scores 87.99 versus LoRA's 89.96, but this gap is small relative to the overall robustness of the method.

On Llama-3-8B, HRHA achieves an average accuracy of 75.35, ranking second only to PoLAR (75.49) and ahead of LoRA (74.03), GraLoRA (73.70), LoHa (73.31), HiRA (70.84), and SVFT (70.09). More importantly, HRHA delivers the best performance on MultiArith (97.83) and SingleEq (95.28), with gains of 2.00 and 0.59 points over the second-best methods, respectively. On gsm8k, HRHA attains 63.68, closely trailing PoLAR's best of 64.29, and on SVAMP it reaches 76.40, just 0.30 points behind PoLAR. This indicates that HRHA is particularly effective for tasks requiring precise symbolic manipulation and multi-step reasoning.

The parameter efficiency of HRHA is a decisive advantage. On Llama-2-7B, it uses only 4.19M trainable parameters, which is about half of LoRA, LoHa, HiRA, GraLoRA, and PoLAR (8.39M), and even slightly less than SVFT (4.45M). On Llama-3-8B, HRHA requires just 2.62M parameters, which is about $2.6\times$ fewer than the rank-based baselines (6.82M) and lower than SVFT (2.78M). Despite this compactness, HRHA remains among the top two methods on both models, demonstrating that its full-rank block-wise parameterization achieves high accuracy without excessive parameter consumption.

The results also reveal significant instability among the higher-rank and compressed baselines. PoLAR performs best on Llama-3-8B but drops to fourth on Llama-2-7B, indicating model-dependent behavior. HiRA performs poorly on Llama-2-7B (47.67) and remains weak on Llama-3-8B (70.84), suggesting that its high-rank construction is not effectively optimized for arithmetic tasks. GraLoRA is competitive on Llama-2-7B but loses to HRHA on Llama-3-8B despite using more parameters. In contrast, HRHA consistently ranks among the top two methods on both model families, with particular strengths in MultiArith and SingleEq, highlighting its robust generalization across model scales.

Overall, these results confirm that HRHA offers an excellent balance between expressive capacity and parameter efficiency for arithmetic reasoning, delivering state-of-the-art or near-state-of-the-art performance while using significantly fewer trainable parameters than existing PEFT methods.

D.7 Ablation Studies

Block Size Table 6 presents the ablation results on block size for HRHA on image classification, evaluated on ViT-Large across seven datasets. The block size directly controls the granularity of the block partition and, consequently, the number of trainable parameters. As the block size decreases from 256×256 to 32×32, the parameter count increases from 196K to 393K, 786K, and 1572K, respectively, consistent with the theoretical $O(mn/b)$ scaling.

The average accuracy improves monotonically as the block size decreases: 88.28 at 256×256, 89.58 at 128×128, 90.61 at 64×64, and 91.10 at 32×32. This confirms that finer block partitions enable the model to capture more localized and fine-grained adaptations, leading to higher representational capacity. However, the performance gains diminish rapidly: the improvement from 256 to 128 is 1.30 points, from 128 to 64 is 1.03 points, and from 64 to 32 is only 0.49 points, while the parameter count doubles at each step.

The benefit of smaller block sizes is most pronounced on challenging fine-grained and structured datasets. On StanfordCars, accuracy rises from 81.75 to 85.98 as the block size decreases, and on FGVC it jumps from 56.55 to 70.12, an improvement of 13.57 points. On OxfordPets and RESISC45, smaller blocks also yield consistent but more modest gains. In contrast, on CIFAR10 and EuroSAT, which are already near saturation, the block size has negligible effect: CIFAR10 remains around 99.06–99.12, and EuroSAT fluctuates within 98.77–99.04. This indicates that finer blocks are particularly valuable when the task demands high-resolution spatial or semantic discrimination, whereas easy tasks do not benefit from the additional capacity.

Although the smallest block size 32×32 achieves the highest average accuracy, it requires 1572K parameters, which is $8\times$ more than the 256×256 configuration and $4\times$ more than 128×128. The marginal gain from 128×128 to 32×32 is only 1.52 points, while the parameter overhead is 1179K. By contrast, moving from 256×256 to 128×128 yields a substantial 1.30-point improvement for only 197K additional parameters. This makes 128×128 the most cost-effective setting: it captures most of the benefit of finer partitioning while maintaining a compact parameter budget. We therefore adopt 128×128 as the default block size for image classification tasks.

Overall, the ablation study demonstrates a clear trade-off between block granularity and parameter efficiency, and validates that a moderate block size provides the best balance between accuracy and computational cost.

Initialization of U, V Table 7 reports the ablation results on the initialization of the shared singular matrices U and V for HRHA on the GLUE benchmark, evaluated on $\text{RoBERTa}_{\text{base}}$ and $\text{RoBERTa}_{\text{large}}$. We compare three strategies: joint bi-sided PCA, random orthogonal initialization, and uniform (non-orthogonal) initialization.

The most striking observation is the severe degradation caused by uniform initialization. On $\text{RoBERTa}_{\text{base}}$, the average score drops from 82.6 (both orthogonal variants) to 61.1, a loss of more than 21 points. The degradation is particularly catastrophic on CoLA, where the score falls to 26.7, and on STS-B, where it drops to 60.1, indicating that the model largely fails to learn meaningful representations. On $\text{RoBERTa}_{\text{large}}$, the effect is even more pronounced: the average score collapses from 86.5/86.7 to 54.4, with CoLA reaching only 9.2 and STS-B 45.0, both near or below random performance. These results confirm that maintaining orthogonality of U and V is essential for the representational capacity and training stability of HRHA. Without orthogonality, the invertibility and rank-preserving properties of the parameterization are destroyed, leading to severe optimization difficulties and poor generalization.

When orthogonality is preserved, the choice between joint bi-sided PCA and random orthogonal initialization has only a marginal effect. On $\text{RoBERTa}_{\text{base}}$, both methods achieve an identical average score of 82.6, with per-task differences within the range of random variation: for example, on MRPC, PCA achieves 87.7 while random orthogonal achieves 87.6, and on RTE, random orthogonal slightly outperforms PCA (72.1 vs. 71.5). On $\text{RoBERTa}_{\text{large}}$, random orthogonal initialization

attains an average of 86.7, slightly higher than PCA's 86.5, and shows small gains on MRPC (88.5 vs. 88.2), CoLA (66.0 vs. 65.1), and RTE (84.3 vs. 84.0). These differences are not statistically significant and suggest that once orthogonality is enforced, the specific orthogonal bases has little impact on downstream performance. This is an encouraging property, as it indicates that HRHA does not rely heavily on a carefully crafted initialization and can achieve strong results with a simple random orthogonal bases.

The ablation study clearly demonstrates that the orthogonality constraint on the shared singular matrices is the critical factor for the success of HRHA. The method is robust to the particular choice of orthogonal initialization, but fails entirely when orthogonality is violated. This finding aligns with the theoretical analysis, which relies on the invertibility of U and V to preserve full rank and expressive capacity. In practice, either joint bi-sided PCA or random orthogonal initialization can be used, with the latter offering a simpler and equally effective option.