SMoA: Spectrum Modulation Adapter for Parameter-Efficient Fine-Tuning

Yongkang Liu[1] **Xing Li**[1] **Mengjie Zhao**[1*] **Shanru Zhang**[1] **Zijing Wang**[1]
Qian Li[2] **Shi Feng**[1] **Feiliang Ren**[1] **Daling Wang**[1] **and Hinrich Schütze**[3,4]
[1]Northeastern University, China; [2]Shandong University, China;
[3]CIS, LMU Munich, Germany;[4]MCML, Germany

Abstract

As the number of model parameters increases, parameter-efficient fine-tuning (PEFT) has become the go-to choice for tailoring pre-trained large language models. Low-rank Adaptation (LoRA) uses a low-rank update method to simulate full parameter fine-tuning, which is widely used to reduce resource requirements. However, decreasing the rank encounters challenges with limited representational capacity. Theory suggests that LoRA fine-tuning with rank r converges toward the top r singular values of the pre-trained weight matrix. As the rank increases, more principal singular directions are preserved, which generally improves the model's performance. However, a larger rank also introduces more trainable parameters, leading to higher computational cost. To overcome this dilemma, we propose SMoA, a **S**pectrum **Mo**dulation **A**dapter that enlarges the accessible family of spectrum-aware updates under a smaller parameter budget. SMoA partitions the layer into multiple aligned spectral blocks and applies one in-block Hadamard-modulated low-rank branch to each diagonal block, yielding broader coverage of pretrained spectral directions. We provide theoretical analysis and empirical results on multiple tasks. In our experiments, SMoA improves average performance in the current lower-budget setting over LoRA and competitive LoRA-style baselines.

1 Introduction

Large language models (LLMs) [3, 26, 49, 2] have demonstrated remarkable performance improvements across a wide range of natural language processing tasks. Fine-tuning (FT) is a standard approach for adapting LLMs to specific downstream tasks [35, 42, 30]. However, the sheer scale of modern LLMs makes full-parameter FT prohibitively expensive, especially in resource-constrained environments [14, 20]. Parameter-Efficient Fine-Tuning (PEFT) is introduced to reduce this cost [15, 23, 22, 29, 16, 13]. PEFT lowers the memory and optimization burden by restricting training to a small set of additional parameters.

Low-rank adaptation [16] is widely used because it introduces very little additional memory overhead and no extra inference latency. However, previous studies [19, 27, 51] have shown that LoRA and most of its variants [25, 13] can struggle on complex tasks such as mathematical reasoning and learning new knowledge [30]. Since a rank-r LoRA update is restricted to rank at most r, adaptation is confined to a limited set of global spectral directions [38]. Gradient-flow theory shows that, for low-rank approximation objectives, rank-r LoRA converges to the optimal rank-r truncated SVD solution, so the residual error is governed by the singular-value tail beyond rank r [38]. Meanwhile, random-matrix analyses suggest that transformer weights often have long-tailed spectra and numerical rank close to $\min(d_{\text{out}}, d_{\text{in}})$, with even small singular directions remaining functionally relevant [40]. These results suggest a structural limitation of a single global LoRA branch: when useful signal

*Corresponding authors.

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

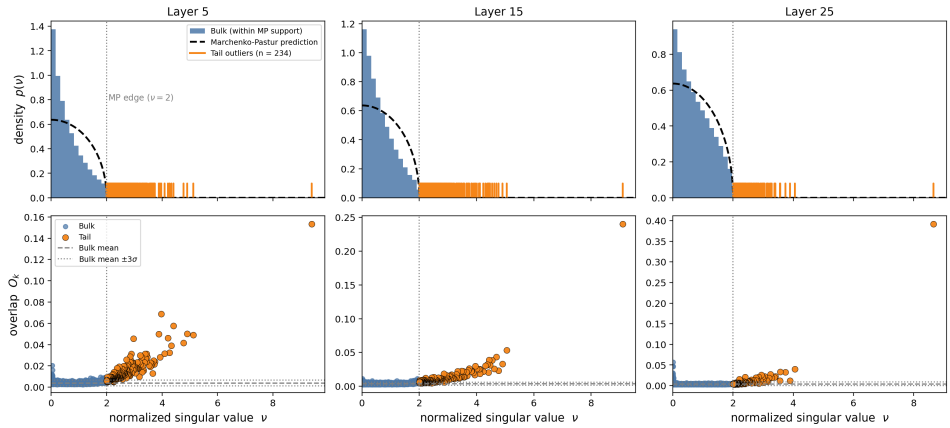

Figure 1: **Pretrained weights have informative spectral tails.** Top: Distribution of normalized singular values ν for Llama-2-7B's layer 5, 15, and 25. The black dashed curve shows Marchenko–Pastur prediction for a random matrix of the same shape; values beyond the bulk edge (vertical line) are tail outliers encoding structured and task-relevant information. Bottom: Maximum overlap $\mathcal{O}_k$ between right singular vectors and activation-covariance eigenvectors. Tail singular directions (orange) still show substantially larger $\mathcal{O}_k$, indicating alignment with task-induced activation structure. Tail directions are therefore not noise but carry functionally relevant signal.

persists in the spectral tail, restricting the update to rank r leaves potentially informative directions under-modeled.

A natural response is to enlarge the accessible update family beyond a single global low-rank branch. Existing methods such as ReLoRA [24], COLA [44], MELoRA [37], and HiRA [18] move in this direction by stacking, composing, or modulating LoRA-style updates. However, they do not explicitly organize adaptation around the spectral structure of the pretrained operator, which may limit their ability to capture informative tail directions under a small parameter budget.

In this paper, we propose a **S**pectrum **Mo**dulation **A**dapter (SMoA) that explicitly targets this spectral regime. SMoA partitions both the output and input dimensions into K aligned spectral blocks, applies one Hadamard-modulated low-rank branch to each diagonal block, and combines the resulting local updates into one global block-diagonal residual. This construction distributes the PEFT budget across K spectrum-aware anchors, allowing the update to interact with a broader collection of singular directions than a single global rank-r factorization. Under near-full-rank block structure and $K \ll \min(d_{\text{out}}, d_{\text{in}})$, our analysis shows that SMoA admits a strictly larger rank ceiling than rank-r LoRA. We evaluate SMoA on Llama-2-7B and Llama-3-8B across commonsense reasoning, dialogue generation, and mathematical reasoning tasks. The results suggest stronger average performance in the lower-budget setting considered here.

We summarize our contributions as follows: **(1)** We propose SMoA, which partitions both dimensions of a pretrained operator into aligned spectral blocks and applies one Hadamard-modulated low-rank branch of local rank r/K to each diagonal block. **(2)** We theoretically characterize the parameter complexity and rank ceiling of SMoA, show that under near-full-rank block structure its rank ceiling is strictly larger than that of rank-r LoRA, identify a family of block-aligned modulated targets that are representable by SMoA but not by rank-r LoRA, and show that on this witness family rank-r LoRA suffers an irreducible spectral approximation gap while SMoA attains zero error. **(3)** Experimental results suggest that SMoA attains stronger average performance than LoRA and remains competitive with strong LoRA variants across multiple benchmarks in the lower-budget setting studied here.

2 Related Work

The scale of modern large language models (LLMs) has made parameter-efficient fine-tuning (PEFT) a central research topic [9, 12]. PEFT aims to approach the performance of full fine-tuning while updating only a small subset of model parameters [30]. Existing methods are commonly grouped into prompt-based [23, 28], adapter-based [15], selection-based [46], and low-rank adaptation approaches [48], among which low-rank adaptation has become one of the dominant paradigms.

LoRA [16] represents weight updates as the product of two low-rank matrices, achieving strong efficiency without increasing inference cost. Building on this framework, VeRA [21] learns only scaling vectors on top of frozen random bases, DoRA [27] decouples direction and magnitude in the update, FLoRA [43] introduces Hadamard-product-based example-specific adaptation, and SSMLoRA [45] connects low-rank adapters across layers via a state-space mechanism. PiSSA [33] is a parameter-efficient fine-tuning method that adapts the principal singular values and singular vectors of pretrained weight matrices.

Another line of work aims to improve the expressive capacity of low-rank adaptation. MoRA [19] replaces the standard low-rank update with a square higher-rank matrix. MELoRA [37] ensembles multiple small low-rank adapters, while HiRA [18] employs element-wise products to realize effectively higher-rank task-specific updates. CoTo [52] improves adaptation by progressively expanding adapter activation during training, encouraging broader exploration of the optimization landscape.

Our work is closest in spirit to methods that seek to overcome the limitations of a single global low-rank branch, but differs in both motivation and construction. Rather than increasing rank through more complex parameterizations or combining multiple LoRA modules, SMoA is motivated by the spectral structure of pretrained Transformer weights, which are often long-tailed and numerically near full-rank. Accordingly, SMoA performs spectrum-aware reordering and allocates adaptation across multiple aligned local subspaces. Each local branch is anchored by a pretrained spectral block and modulated via a Hadamard product, enabling rank accumulation across blocks under a structured spectrum-aware inductive bias. This makes SMoA a spectrum-aware alternative to standard LoRA-style global low-rank adaptation.

3 Methodology

We first explain why pretrained transformer weights are better viewed as numerically near full-rank matrices with informative spectral tails than as effectively low-rank operators. This motivates an adapter that can preserve or modulate more than only a few leading singular directions. We then construct SMoA by distributing a small PEFT budget across aligned spectral blocks of the frozen weight, and finally analyze its parameter count, potential rank ceiling, and expressivity relative to a rank-r LoRA layer.

3.1 Spectral Modulation

We illustrate the informative spectral tails of pretrained model weights in Figure 1, following the random matrix-based analysis of Transformer models [40]. Figure 1 top images highlight that a pretrained weight's singular-value spectrum has a heavy tail: a substantial fraction of singular values lies beyond the Marchenko-Pastur bulk [32] predicted for a same-size random matrix, confirming that the pretrained weight indeed encodes task-relevant information in the spectrum tail. Bottom images further confirm that directions on the tail are not random noise and they have right singular vectors aligning more (exceeding 3σ intervals) with the activation-covariance eigenvectors (see below) than the bulk directions. Overall, these observations indicate that pretrained weights are numerically near full-rank with informative tails, consistent with prior random-matrix analyses [40].

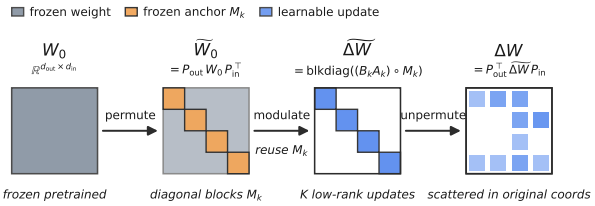

Figure 2: Overview of SMoA pipeline (when $K = 4$).

We record above empirical regularity as Lemma 1 (Appendix G.1). If informative directions extend into the spectral tail, restricting the update to a rank-r subspace leaves potentially useful directions under-modeled. Motivated by this, SMoA is designed to preserve and exploit a broader range of spectrum-aware structure than a single global low-rank branch. For a frozen linear layer $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, we construct SMoA in three steps shown in Figure 2: (i) reorder the coordinates once according to the frozen spectral structure of W_0, (ii) build one Hadamard-modulated low-rank branch on each aligned diagonal block, and (iii) map the resulting block-diagonal update back to the original

coordinates. Additional preprocessing details appear in Appendix E.1, while the deferred theoretical statements and proofs appear in Appendix G.3, Appendix G.4, and Appendix G.5.

Spectrum-aware preprocessing. Let $W_0 = U\Sigma V^\top$ be the singular value decomposition of the pretrained weight. We use this frozen spectral structure only once to define output and input permutations, denoted by π_{out} and π_{in}, that reorder the coordinates into K spectrum-aligned contiguous groups. The corresponding permutation matrices are

$$(P_{\text{out}})_{ij} = \mathbf{1}[j = \pi_{\text{out}}(i)] \text{ for } i, j \in \{1, \dots, d_{\text{out}}\}, \quad (P_{\text{in}})_{ij} = \mathbf{1}[j = \pi_{\text{in}}(i)] \text{ for } i, j \in \{1, \dots, d_{\text{in}}\}. \tag{1}$$

The reordered weight matrix $\widetilde{W}_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ is then

$$\widetilde{W}_0 = P_{\text{out}} W_0 P_{\text{in}}^\top, \qquad (\widetilde{W}_0)_{ij} = (W_0)_{\pi_{\text{out}}(i),\, \pi_{\text{in}}(j)} \text{ for } i \in \{1, \dots, d_{\text{out}}\},\ j \in \{1, \dots, d_{\text{in}}\}. \tag{2}$$

This reordering leaves the layer computation unchanged; it only exposes a coordinate system in which related spectral coordinates become contiguous.

Block anchors. SMoA places its local branches on aligned diagonal blocks of $\widetilde{W}_0$. For exposition, assume that K divides both d_{out} and d_{in}, and let $\mathcal{O}_k$ and $\mathcal{I}_k$ denote the k-th contiguous row and column intervals after reordering, each of size d_{out}/K and d_{in}/K, respectively. The k-th diagonal anchor is then $M_k = \widetilde{W}_0[\mathcal{O}_k, \mathcal{I}_k] \in \mathbb{R}^{\frac{d_{\text{out}}}{K} \times \frac{d_{\text{in}}}{K}}$. (see orange blocks in Figure 2). We refer to $\{M_k\}_{k=1}^K$ as the reordered block anchors of W_0. These anchors are fixed after preprocessing and are not updated during fine-tuning. Note that $\widetilde{W}_0$ itself need not be block diagonal; SMoA only uses its aligned diagonal blocks as local spectral anchors. Additional indexing details are deferred to Appendix E.1.

Adapter update. Let the reference LoRA baseline use a global rank budget r. To preserve the same total rank budget while distributing adaptation capacity across K spectrum-aligned blocks, we define the **local block rank** as $\rho = \frac{r}{K}$. SMoA then attaches one local branch of rank ρ to each diagonal anchor block. Concretely, for each block M_k, we introduce learnable low-rank factors $A_k \in \mathbb{R}^{\rho \times \frac{d_{\text{in}}}{K}}$, $B_k \in \mathbb{R}^{\frac{d_{\text{out}}}{K} \times \rho}$, and define the corresponding Hadamard-modulated local update by $\Delta M_k = (B_k A_k) \odot M_k$. The full update in reordered coordinates is

$$\widetilde{\Delta W} = \text{blkdiag}(\Delta M_1, \dots, \Delta M_K) = \text{blkdiag}\left((B_1 A_1) \odot M_1, \dots, (B_K A_K) \odot M_K\right). \tag{3}$$

Scattering this update back to the original coordinates yields

$$\Delta W = P_{\text{out}}^\top \widetilde{\Delta W}\, P_{\text{in}}, \qquad W = W_0 + \Delta W. \tag{4}$$

Thus, $(P_{\text{out}}, P_{\text{in}}, \{M_k\}_{k=1}^K)$ remain fixed, while the trainable parameters are $\{A_k, B_k\}_{k=1}^K$. Compared with LoRA, SMoA distributes the adaptation budget across K local spectrum-aligned branches instead of a single global low-rank branch.

3.2 Analysis: Spectrum-Aware Rank Capacity and Expressivity

We next compare SMoA with a standard rank-r LoRA layer in parameter count and achievable rank capacity; fuller witness-family separation results are deferred to Appendix G.4 and Appendix G.5.

Parameter complexity. Compared with the $r(d_{\text{in}} + d_{\text{out}})$ trainable parameters of a rank-r LoRA layer, SMoA with K blocks and local rank $\rho = \frac{r}{K}$ uses

$$P_{\text{SMoA}} = \sum_{k=1}^{K} \left(\rho \frac{d_{\text{in}}}{K} + \rho \frac{d_{\text{out}}}{K}\right) = \frac{r}{K}(d_{\text{in}} + d_{\text{out}}) = \frac{1}{K} r (d_{\text{in}} + d_{\text{out}}), \tag{5}$$

that is, only a $\frac{1}{K}$ fraction of the trainable parameters required by rank-r LoRA. Next, we show the main rank-capacity consequence of SMoA's block-wise construction.

Proposition 1 (Spectrum-Aware Rank Ceiling)**.** *Let*

$$\widetilde{\Delta W} = \text{blkdiag}(\Delta M_1, \dots, \Delta M_K), \qquad \Delta M_k = (B_k A_k) \odot M_k, \qquad \rho = \frac{r}{K}. \tag{6}$$

Here $A_k \in \mathbb{R}^{\rho \times \frac{d_{in}}{K}}$, $B_k \in \mathbb{R}^{\frac{d_{out}}{K} \times \rho}$, $M_k \in \mathbb{R}^{\frac{d_{out}}{K} \times \frac{d_{in}}{K}}$, *and* $s_k = \min\left(\frac{d_{out}}{K}, \frac{d_{in}}{K}\right)$. *Then, for each block,*

$$\operatorname{rank}(\Delta M_k) \le \min(s_k,\ \rho\ \operatorname{rank}(M_k))\,. \tag{7}$$

Consequently, the full SMoA update satisfies

$$\operatorname{rank}(\Delta W) = \operatorname{rank}(\widetilde{\Delta W}) \le U := \sum_{k=1}^{K} \min(s_k,\ \rho\ \operatorname{rank}(M_k)) \le \min(d_{out}, d_{in}). \tag{8}$$

In contrast, a standard rank-r *LoRA update always satisfies* $\operatorname{rank}(\Delta W_{\mathrm{LoRA}}) \le r$. *Therefore*

$$U > r, \tag{9}$$

SMoA admits a strictly larger analytic rank ceiling than rank-r *LoRA.*

Please refer to Appendix G.2 for the proof. Appendix G.3 Corollary 1 also shows a useful special case: under full-rank local anchors, this ceiling is strictly larger than that of a standard rank-r LoRA update. Proposition 2 constructs a block-aligned witness family that is exactly representable by SMoA but can fall outside the rank-r LoRA family when its reordered rank exceeds r. Corollary 2 in Appendix G.5 then shows that, on this same witness family, the best rank-r LoRA approximation incurs the truncated-spectrum residual $\sum_{j>r} \sigma_j(\widetilde{\Delta W}^{\star})^2$, whereas SMoA attains zero error. This is consistent with the gradient-flow theory of Rushka and Klabjan [38]: under Frobenius low-rank approximation objectives and spectral initialization, LoRA converges to the optimal rank-r truncated-SVD solution, so the residual is a structural consequence of the restricted update family rather than merely an optimization artifact.

Taken together, Equation (8), Proposition 1, and the appendix results characterize the expressive advantage of SMoA from three complementary perspectives: a smaller parameter budget via Equation (5), a larger analytic rank ceiling via Proposition 1, and a one-sided witness-family separation via Proposition 2. Appendix G.4 and Appendix G.5 show the detailed witness construction and proofs. Because SMoA remains block diagonal in the reordered coordinates, this advantage is structural rather than universal; if a task residual is dominated by strong cross-block interactions, the same inductive bias can become restrictive.

Table 1: Accuracy comparison among PEFT methods. Results for ChatGPT are sourced from [27]. The best performance within is indicated in **bold**, while the second-best performance is highlighted with underlining. The reported numbers for SMoA are averaged accuracy (standard deviation).

Model	Method	Params(%)	BoolQ	PIQA	SIQA	ARC-c	ARC-e	OBQA	HellaS	WinoG	Average
ChatGPT	-	-	73.10	85.40	68.50	79.90	89.80	74.80	78.50	66.10	77.01
Llama-2-7B	PromptTuning	0.0012	55.93	12.35	30.50	6.06	8.63	9.40	6.91	40.57	21.29
	P-Tuning	0.7428	58.75	36.02	0.20	0.17	1.98	0.80	0.01	0.00	12.24
	LoRA(r=32)	0.8256	69.80	79.90	79.50	64.70	79.80	81.00	83.60	82.60	77.61
	DoRA(r=32)	0.8256	71.80	83.70	76.00	68.20	83.70	82.40	89.10	82.60	79.69
	MoRA(r=32)	0.8241	72.17	80.79	79.53	71.42	85.31	81.20	29.09	80.19	72.46
	SSMLoRA(r=32)	0.8024	68.35	76.57	78.58	64.57	78.85	80.68	82.78	82.38	76.60
	MeLoRA(r=16,K=2)	0.4128	70.12	80.65	75.35	67.67	82.55	80.28	86.05	78.23	77.61
	MeLoRA(r=32,K=2)	0.8256	71.43	82.83	77.65	71.89	85.38	81.35	87.52	81.35	79.93
	HiRA(r=16)	0.4128	69.82	80.20	78.20	71.33	85.90	81.00	86.99	83.43	79.61
	HiRA(r=32)	0.8256	71.22	83.35	79.53	73.81	86.74	**84.60**	88.12	83.98	81.42
	SMoA(r=16,K=2)	0.4128	70.13(0.35)	81.27(0.52)	78.31(0.65)	72.53(0.28)	86.41(0.55)	81.03(0.83)	87.42(0.71)	83.68(0.57)	80.10
	SMoA(r=32,K=2)	0.8256	**72.65**(0.55)	**83.74**(0.43)	**79.85**(0.55)	**74.57**(0.65)	**87.22**(0.62)	82.82(0.47)	89.56(0.58)	**84.38**(0.75)	**81.85**
Llama-3-8B	PromptTuning	0.0010	56.85	45.05	36.13	31.57	32.74	29.20	14.01	50.12	36.96
	P-Tuning	0.6240	59.97	11.64	8.19	7.42	8.63	9.60	1.77	37.65	18.11
	LoRA(r=32)	0.7002	70.80	85.20	79.90	71.20	84.20	79.00	91.70	84.30	80.79
	DoRA(r=32)	0.7002	74.60	89.30	79.90	80.40	90.50	85.80	95.50	85.60	85.20
	MoRA(r=32)	0.6997	74.28	87.43	80.71	79.61	91.16	85.60	43.53	86.74	78.63
	SSMLoRA(r=32)	0.6813	68.57	84.38	77.95	70.81	85.38	78.87	90.50	84.33	80.10
	MeLoRA(r=16,K=2)	0.3513	72.85	87.33	79.32	78.38	88.65	85.63	92.78	84.38	83.67
	MeLoRA(r=32,K=2)	0.7002	74.53	87.35	80.54	78.43	90.37	86.38	93.76	86.37	84.72
	HiRA(r=16)	0.3513	73.85	89.12	81.06	82.59	93.06	87.40	94.85	86.74	86.08
	HiRA(r=32)	0.7002	75.40	**89.70**	81.15	82.90	**93.27**	88.32	95.36	87.70	86.73
	SMoA(r=16,K=2)	0.3513	74.56(0.55)	87.85(0.63)	81.13(0.35)	82.57(0.45)	92.43(0.56)	87.86(0.55)	95.13(0.66)	86.77(0.47)	86.04
	SMoA(r=32,K=2)	0.7002	**76.52**(0.56)	89.24(0.72)	**82.35**(0.55)	**83.65**(0.55)	93.25(0.45)	**89.54**(0.67)	**95.88**(0.57)	**88.72**(0.62)	**87.39**

4 Experiments

We conduct experiments on three task families: commonsense reasoning, dialogue generation, and mathematical reasoning. For commonsense reasoning, we evaluate on eight standard benchmarks: BoolQ [5], PIQA [4], SIQA [39], ARC-c [6], ARC-e [6], OBQA [34], HellaSwag [47], and WinoGrande [1]. For dialogue generation and mathematical reasoning, we use ConvAI2 [10] and GSM8K [8], respectively. We compare SMoA with representative PEFT methods, including prompt-based approaches and recent LoRA variants, and report the standard metrics for each task family. To keep the main text focused, we provide the complete experimental details in the appendix: Appendix A

describes the datasets and task formulations; Appendix B introduces the compared baselines; Appendix C specifies the evaluation metrics and answer-extraction protocols; and Appendix D reports the training setup, implementation choices, and hyperparameter configurations.

Table 2: Performance comparison among various PEFT methods on the CONVAI2 dataset using BLEU, BERTScore F1/Recall/Precision, METEOR, ROUGE-L, and average score.

Model	Method	Params(%)	BLEU	BERT F1	BERT R	BERT P	Meteor	R-L	Average
Llama-2-7B	PromptTuning	0.0012	0.04	72.44	77.38	68.23	0.80	0.80	36.62
	P-Tuning	0.7428	0.60	83.29	83.33	83.28	**15.11**	12.36	46.33
	LoRA(r=32)	0.8256	1.82	84.41	84.71	84.16	11.38	10.55	46.17
	DoRA(r=32)	0.8256	1.73	84.18	84.61	83.81	11.25	10.41	46.00
	MoRA(r=32)	0.8241	1.09	84.09	84.65	83.59	10.97	9.57	45.66
	SSMLoRA(r=32)	0.8024	1.56	84.36	84.28	83.56	10.67	10.44	45.81
	MeLoRA(r=16,K=2)	0.4128	2.46	83.32	83.33	82.72	12.37	10.65	45.81
	MeLoRA(r=32,K=2)	0.8256	2.55	84.77	84.08	84.58	13.27	11.87	46.85
	HiRA(r=16)	0.4128	2.56	83.97	84.12	83.86	13.35	12.58	46.74
	HiRA(r=32)	0.8256	2.70	84.86	84.98	84.77	13.56	12.80	47.28
	SMoA(r=16,K=2)	0.4128	2.58(0.67)	83.25(0.55)	83.53(0.47)	83.37(0.52)	13.15(0.25)	12.35(0.33)	46.42
	SMoA(r=32,K=2)	0.8256	**2.94**(0.75)	**86.35**(0.68)	**85.55**(0.35)	**85.63**(0.45)	14.26(0.17)	**13.65**(0.51)	**48.06**
Llama-3-8B	PromptTuning	0.0010	1.45	82.99	82.99	83.05	14.72	13.13	46.39
	P-Tuning	0.6240	1.50	81.52	81.07	82.01	15.49	13.55	45.86
	LoRA(r=32)	0.7002	2.26	84.32	84.00	84.67	12.51	11.77	46.59
	DoRA(r=32)	0.7002	2.29	84.32	84.06	84.62	12.63	11.78	46.62
	MoRA(r=32)	0.6997	1.60	84.22	84.06	84.43	12.37	11.19	46.31
	SSMLoRA(r=32)	0.6813	2.45	84.37	84.05	84.57	12.36	11.15	46.49
	MeLoRA(r=16,K=2)	0.3513	1.87	82.36	83.55	84.67	12.39	11.54	46.06
	MeLoRA(r=32,K=2)	0.7002	2.87	84.15	**85.65**	85.37	13.77	13.87	47.61
	HiRA(r=16)	0.3513	3.32	84.84	84.41	85.30	14.93	13.94	47.79
	HiRA(r=32)	0.7002	3.41	84.81	84.40	85.25	14.87	14.05	47.80
	SMoA(r=16,K=2)	0.3513	3.37(0.44)	84.55(0.56)	84.67(0.77)	**85.54**(0.57)	15.37(0.67)	13.89(0.77)	47.89
	SMoA(r=32,K=2)	0.7002	**3.77**(0.52)	**85.25**(0.72)	85.39(0.54)	84.78(0.37)	**16.55**(0.52)	**14.78**(0.55)	**48.42**

4.1 Commonsense Reasoning Tasks

Table 1 reports the accuracy comparison of various PEFT methods on a diverse set of commonsense reasoning benchmarks, including BoolQ, PIQA, SIQA, ARC-c, ARC-e, OBQA, HellaSwag, and WinoGrande, evaluated on the Llama-2-7B [41] and Llama-3-8B [11] backbones. Across both backbones, SMoA($r = 32$, $K = 2$) gives the highest average accuracy in Table 1. On Llama-2-7B, SMoA reaches 81.85% on average, compared with 81.42% for HiRA($r = 32$) and 79.93% for MeLoRA($r = 32$, $K = 2$). On Llama-3-8B, SMoA reaches 87.39% on average, compared with 86.73% for HiRA($r = 32$) and 85.20% for DoRA($r = 32$). These gains should be read in the current lower-budget setting, since SMoA uses fewer trainable parameters than rank-r LoRA. These results suggest that SMoA is most useful on tasks that benefit from broader *structured* adaptation.

SMoA also remains competitive under more parameter-efficient settings. With $r = 16$ and $K = 2$, it attains an average accuracy of 80.10% on Llama-2-7B while using substantially fewer trainable parameters than most rank-32 methods, yet still matching or exceeding several baselines. On Llama-3-8B, SMoA with $r = 16$ reaches 86.04% on average. These results suggest a favorable accuracy-efficiency trade-off. Overall, the commonsense reasoning results indicate that SMoA is competitive across model scales and maintains a favorable balance of performance and parameter efficiency. We show that SMoA gives the best average performance in Table 1 while remaining competitive on most individual tasks.

Table 3: Comparison of PEFT methods.

Model	Trainable	Method	GSM8K
Llama-3-8B	0.0012	PromptTuning	15.62
	0.7428	P-Tuning	2.65
	0.8256	LoRA(r=32)	65.89
	0.8256	DoRA(r=32)	66.12
	0.8241	MoRA(r=32)	67.89
	0.8024	SSMLoRA(r=32)	64.75
	0.4128	MeLoRA(r=16,K=2)	66.76
	0.8256	MeLoRA(r=32,K=2)	69.36
	0.4128	HiRA(r=16)	67.63
	0.8256	HiRA(r=32)	70.81
	0.4128	SMoA(r=16,K=2)	68.55(0.73)
	0.8256	SMoA(r=32,K=2)	**72.33**(0.86)

4.2 Conversational Task

Table 2 reports a comprehensive comparison of various PEFT methods on the CONVAI2 dataset under two backbone models, Llama-2-7B and Llama-3-8B. Across both backbones, SMoA ($r = 32$, $K = 2$) yields the highest Average score, reaching 48.06% on Llama-2-7B and 48.42% on Llama-3-8B. This represents a clear improvement over strong baselines such as LoRA, HiRA, and MeLoRA with comparable parameter budgets. Overall, the results demonstrate that SMoA consistently achieves the best or near-best performance across most automatic evaluation metrics, validating its effectiveness for dialogue generation. Compared to LoRA and its variants (DoRA, MoRA, SSMLoRA), SMoA consistently delivers higher BLEU, METEOR, and ROUGE-L scores. This suggests that SMoA is more effective at capturing both surface-level n-gram overlap and longer-range semantic coherence. Relative to HiRA and MeLoRA, which introduce hierarchical or multi-expert structures, SMoA

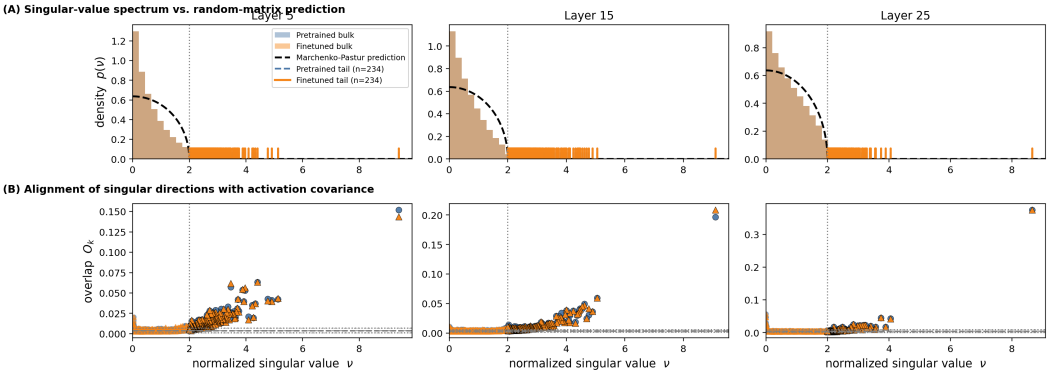

Figure 3: **Original Llama2-7B vs. SMoA finetuned spectra and overlaps on BoolQ for key weight matrix of attention.** We compare all layers before and after finetuning. The figure shows the 5th, 15th, and 25th layers, results for other layers are in the Appendix F. Top: empirical distributions of normalized singular values ν; light brown denotes the pretrained model and orange denotes the finetuned model. The dashed curve is the theoretical random-matrix bulk. Bottom: the corresponding maximum overlap O_k between right singular vectors and activation-covariance eigenvectors, with circles for the pretrained model and triangles for the finetuned model. Gray lines indicate the bulk mean and $\pm 3\sigma$ intervals. The figure highlights how finetuning changes the right-tail spectrum and the alignment of tail singular directions with task-induced activation structure.

demonstrates superior performance across metrics. For example, although HiRA achieves strong BERT F1 scores, SMoA surpasses it in BLEU and METEOR, reflecting better lexical diversity and relevance in generated responses. This suggests that SMoA's multidimensional space adapter approach provides a more balanced trade-off between semantic fidelity and generation quality.

4.3 Mathematical Reasoning Task

We evaluate SMoA on GSM8K using Llama-3-8B as the backbone model. As shown in Table 3, SMoA reaches 72.33% accuracy, compared with 65.89% for LoRA, 66.12% for DoRA, and 67.89% for MoRA. Even under a smaller parameter budget ($r = 16$, $K = 2$), SMoA remains competitive at 68.55%. These results suggest that SMoA adapts well to challenging tasks under limited resources.

Compared with LoRA and related methods, SMoA offers a useful balance between parameter efficiency and broader spectrum-aware adaptation, which may be particularly relevant for mathematical reasoning. Its performance is consistent with the view that broader spectrum-aware adaptation can enhance the model's capacity for mathematical reasoning. This observation should again be interpreted as task-dependent: the benefit comes from broader spectrum-aware adaptation, not from assuming that higher rank alone is always sufficient.

4.4 Spectral Structure Comparison

Figure 3 reports three representative layers, while the complete layer-wise visualization is provided in Appendix F.1. A clear and consistent pattern emerges: compared with the pretrained model, whose spectrum is mostly confined to the random-matrix bulk, SMoA fine-tuning produces a pronounced spectral component beyond the bulk edge. This right-tail shift suggests that SMoA does not simply perturb dominant pretrained directions, but selectively activates underutilized singular directions that are likely to encode task-specific information.

The corresponding overlap scores provide functional evidence for this interpretation. Before fine-tuning, the singular directions exhibit weak alignment with the activation-covariance eigenspace, with most O_k values close to the bulk baseline. After SMoA fine-tuning, large overlaps appear mainly in the intermediate and tail spectral regions, indicating that the directions emphasized by SMoA are also more aligned with task-induced activation structure. Therefore, the spectral shift is not a random by-product of optimization, but reflects a structured adaptation mechanism.

These results highlight the advantage of SMoA: it preserves the pretrained spectral bulk while reallocating adaptation capacity to a small number of informative tail directions. This provides an explanation for why SMoA can achieve effective fine-tuning with limited trainable parameters. The

Table 4: Tail spectrum effectiveness ablation experiment. Ablation is performed only on the tail spectrum of the incremental weights ΔW.

Ablation	BoolQ	PIQA	SIQA	ARC-c	ARC-e	OBQA	HellaS	WinoG	Average
SMoA	**72.65**	**83.74**	**79.85**	**74.57**	**87.22**	**82.82**	**89.56**	**84.38**	**81.85**
-tail spectrum Q	72.62	83.68	79.80	74.48	87.18	82.75	89.50	84.29	81.79
-tail spectrum K	72.60	83.72	79.82	74.55	87.20	82.78	89.52	84.33	81.82
-tail spectrum V	72.55	83.65	79.75	74.43	87.15	82.75	89.45	84.28	81.75
-tail spectrum QKV	70.35	80.69	76.65	73.36	85.38	80.06	87.37	82.58	79.56

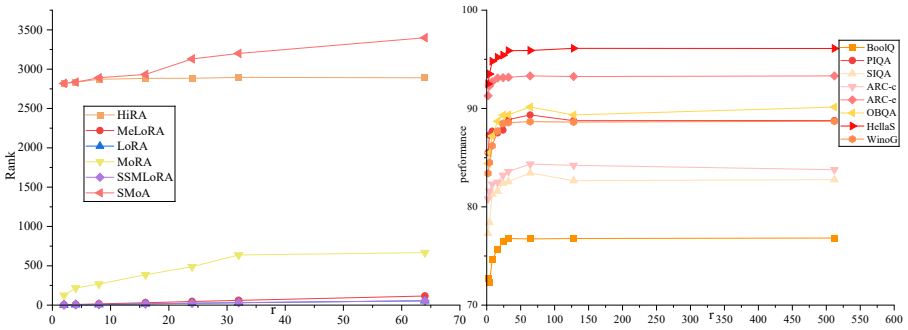

Figure 4: Left: comparison of the measured update rank of ΔW for different PEFT methods after training. For MeLoRA and SMoA, the number of branches is set to 2. SMoA should be interpreted here as the lower-budget Hadamard-modulated block adapter defined in Section 3.2. Right: performance of SMoA across tasks as r increases with $K = 2$.

same trend is observed across all layers in Appendix F.1, demonstrating that the phenomenon is systematic rather than specific to the selected layers.

4.5 Effectiveness of Tail Spectrum

To further verify whether the tail singular spectrum contributes to the capability of SMoA, we conduct an ablation study by removing the tail-spectrum components from different attention projections. As shown in Table 4, removing the tail spectrum from a single projection matrix, i.e., Q, K, or V, consistently leads to performance degradation across most benchmarks. The trend is highly consistent: the full SMoA model achieves the best or near-best performance on all tasks, while $-$*tail spectrum* Q, $-$*tail spectrum* K, and $-$*tail spectrum* V each slightly weaken the model.

More importantly, jointly removing the tail-spectrum components from Q, K, and V causes a much larger degradation. For example, the average performance drops from 81.85% for SMoA to 79.56% after removing the tail spectrum from all three projections. This suggests that the tail spectrum is not merely redundant noise, but instead contains useful task-adaptive information that is distributed across multiple attention projections. These results provide direct empirical evidence for our motivation: effective fine-tuning should not only exploit the dominant singular directions, but also preserve and adapt informative tail-spectrum components. The much stronger degradation under the joint QKV ablation further indicates that the tail spectra of different projections play complementary roles in attention adaptation. Therefore, the superior performance of SMoA can be attributed, at least in part, to its ability to capture and utilize these otherwise overlooked tail-spectrum directions.

4.6 Rank Analysis

We compare the average rank of the update parameter matrix ΔW for various PEFT methods, including HiRA, LoRA, MoRA, and SMoA, all of which employ different strategies for adapting to new tasks. As shown in Figure 4 (left), SMoA attains significantly higher ranks for ΔW across all values of r, indicating its superior ability to perform high-rank adaptation. Compared to LoRA, MoRA, and MeLoRA, SMoA's rank increases dramatically as r increases, suggesting that it is more efficient in capturing task-specific information through higher-rank updates. Notably, the rank of ΔW for HiRA is also considerably higher than that for LoRA and MoRA, though it does not reach the levels of SMoA. This result highlights HiRA's effectiveness in improving the expressiveness of the model while maintaining a more parameter-efficient approach. Both HiRA and SMoA demonstrate that higher-rank updates correlate with improved performance. In summary, the results emphasize the importance of high-rank adaptation in PEFT methods, with SMoA leading in this aspect, achieving substantial improvements over baseline methods, and offering a promising direction for enhancing model performance.

Table 5: Performance on commonsense reasoning datasets under different SMoA block configurations (r, K), using the same metrics as in Table 1. Here r denotes the reference LoRA rank and each SMoA block uses local rank r/K. Boldface indicates the best result for each metric.

Method	$r \times K$	Params(%)	BoolQ	PIQA	SIQA	ARC-c	ARC-e	OBQA	HellaS	WinoG	Average
LoRA	16×1	0.3513	74.44	87.46	80.50	82.15	91.56	87.13	94.62	86.25	85.51
SMoA	8×2	0.1756	**74.60**	87.70	81.30	82.30	**92.70**	87.20	**94.80**	86.20	85.85
SMoA	4×4	0.0878	74.32	87.56	81.24	82.19	92.62	87.13	94.62	86.24	85.74
SMoA	2×8	0.0439	74.52	**87.74**	**81.45**	**82.37**	92.62	**87.24**	94.67	**86.34**	**85.87**
SMoA	1×16	0.0220	74.30	87.25	81.29	82.24	92.25	87.08	94.68	86.28	85.67
LoRA	32×1	0.7002	**74.77**	87.87	81.36	82.59	92.18	88.23	94.88	87.17	86.13
SMoA	16×2	0.3513	74.56	87.85	81.13	82.57	92.43	87.86	95.13	86.77	86.04
SMoA	8×4	0.1757	74.43	**88.34**	81.35	82.73	92.27	88.34	95.17	**87.37**	86.25
SMoA	4×8	0.0878	74.36	88.28	**81.42**	**82.75**	**92.44**	**88.37**	**95.22**	87.32	**86.27**
SMoA	2×16	0.0439	74.28	88.18	81.25	82.62	92.34	88.23	95.02	87.13	86.13
SMoA	1×32	0.0220	74.30	88.22	81.35	82.63	92.23	88.25	95.14	87.16	86.16

4.7 Effect of Rank on Performance

Figure 4 (right) demonstrates how the performance of SMoA changes with respect to the rank parameter r across several commonsense reasoning tasks. As r increases from 2 to 32, we observe a consistent and notable improvement in the model's performance, with average accuracy rising from 83.49% to 87.35%. This trend highlights the impact of increasing the rank r on the model's ability to generalize across diverse reasoning tasks. In particular, tasks such as SIQA, OBQA and WinoG exhibit the most significant performance gains as r increases. SIQA, for example, improves by 5.26% as r rises, emphasizing the importance of a higher rank structure for tasks that demand more sophisticated reasoning capabilities. Notably, even when $r = 8$, SMoA demonstrates a competitive edge, suggesting that it efficiently leverages a smaller number of trainable parameters while maintaining strong performance, making SMoA a compelling choice even under constrained resources.

4.8 Analysis of Block Configurations

In this section, we analyze how the SMoA block configuration affects performance across different tasks. We evaluate several choices of the pair (r, K), where r is the reference LoRA rank and each block uses local rank r/K, and report the results in Table 5.

We make two observations. First, SMoA remains competitive across a wide range of block configurations. In Table 5, several (r, K) choices attain strong accuracy while sometimes using more than $2\times$ fewer trainable parameters than stronger-budget alternatives. This suggests that, in our setting, performance depends not only on the nominal reference rank r, but also on how that rank budget is distributed across spectral blocks. The configuration $(r, K) = (2, 8)$ stands out as one favorable trade-off between accuracy and parameter efficiency: it reaches an average score of 85.87% while keeping the trainable parameter count small. This highlights the importance of choosing a block configuration that balances structural coverage and budget. Ideally, the pair (r, K) should be tuned for each dataset, although $K = 2$ already works well in most cases. Second, the optimal configuration varies across tasks. On ARC-e and HellaSwag, the best performance is usually obtained with $K = 2$, whereas other tasks may benefit from larger values such as $K = 4$ or $K = 8$. We conjecture that this variation is mainly driven by model scale and task complexity, together with how well the reordered block structure matches the downstream residual.

5 Conclusion

In this paper, we propose SMoA, a parameter-efficient fine-tuning method that partitions a pretrained operator into multiple aligned spectral blocks and applies one Hadamard-modulated low-rank branch to each diagonal block. We show that SMoA uses fewer trainable parameters than rank-r LoRA while enlarging the accessible family of spectrum-aware updates through block-diagonal modulation, and we prove both a one-sided structural separation and a spectral approximation gap on a family of block-aligned modulated targets. This connects our method directly to recent LoRA rank theory: rather than only increasing nominal rank, SMoA changes the admissible update family so that some residual directions that would remain in the LoRA spectral tail can be represented exactly. Empirically, SMoA yields stronger average performance together with higher measured update ranks in the lower-budget setting considered here.

Acknowledgments and Disclosure of Funding

This work is supported by the National Natural Science Foundation of China (Grant No.62502081, No.62676083, No.62402293, No.62506216), and the Fundamental Research Funds for the Central Universities (Grant No. N2523011).

References

[1] Winogrande: An adversarial winograd schema challenge at scale. 2019.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[4] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[5] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

[6] 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.

[7] Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168, 2021. URL https://api.semanticscholar.org/CorpusID:239998651.

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

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

[10] 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*, pages 187–208. Springer, 2019.

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

[12] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

[13] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*, 2024.

[14] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 30016–30030, 2022.

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

[16] 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.

[17] 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*, pages 5254–5276, 2023.

[18] 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.

[19] Ting Jiang, Shaohan Huang, Shengyue Luo, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, et al. Mora: High-rank updating for parameter-efficient fine-tuning. *arXiv preprint arXiv:2405.12130*, 2024.

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

[21] Dawid J Kopiczko, Tijmen Blankevoort, and Yuki M Asano. Vera: Vector-based random matrix adaptation. *arXiv preprint arXiv:2310.11454*, 2023.

[22] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021.

[23] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.

[24] Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. Stack more layers differently: High-rank training through low-rank updates. 2023.

[25] Vladislav Lialin, Namrata Shivagunde, Sherin Muckatira, and Anna Rumshisky. Relora: High-rank training through low-rank updates. *arXiv preprint arXiv:2307.05695*, 2023.

[26] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[27] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, pages 32100–32121, 2024.

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

[29] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *AI Open*, 5:208–215, 2024.

[30] Yongkang Liu, Xingle Xu, Ercong Nie, Zijing Wang, Shi Feng, Daling Wang, Qian Li, and Hinrich Schütze. Look within or look beyond? a theoretical comparison between parameter-efficient and full fine-tuning. *arXiv preprint arXiv:2505.22355*, 2025.

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[32] Vladimir A Marčenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483, 1967.

[33] 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.

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

[35] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[36] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[37] Pengjie Ren, Chengshun Shi, Shiguang Wu, Mengqi Zhang, Zhaochun Ren, Maarten Rijke, Zhumin Chen, and Jiahuan Pei. Melora: Mini-ensemble low-rank adapters for parameter-efficient fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3052–3064, 2024.

[38] Michael Rushka and Diego Klabjan. Rank-accuracy trade-off for lora: A gradient-flow analysis. *arXiv preprint arXiv:2602.10212*, 2026.

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

[40] Max Staats, Matthias Thamm, and Bernd Rosenow. Small singular values matter: A random matrix analysis of transformer models. *arXiv preprint arXiv:2410.17770*, 2024.

[41] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[42] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

[43] Yeming Wen and Swarat Chaudhuri. Batched low-rank adaptation of foundation models. In *The Twelfth International Conference on Learning Representations*.

[44] Wenhan Xia, Chengwei Qin, and Elad Hazan. Chain of lora: Efficient fine-tuning of language models via residual learning. *arXiv preprint arXiv:2401.04151*, 2024.

[45] Jiayang Yu, Yihang Zhang, Bin Wang, Peiqin Lin, Yongkang Liu, and Shi Feng. Ssmlora: Enhancing low-rank adaptation with state space model. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4493–4506, 2025.

[46] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, 2022.

[47] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[48] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

[49] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

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

[51] Zhan Zhuang, Yulong Zhang, Xuehao Wang, Jiangang Lu, Ying Wei, and Yu Zhang. Time-varying lora: Towards effective cross-domain fine-tuning of diffusion models. *Advances in Neural Information Processing Systems*, 37:73920–73951, 2024.

[52] Zhan Zhuang, Xiequn Wang, Wei Li, Yulong Zhang, Qiushi Huang, Shuhao Chen, Xuehao Wang, Yanbin Wei, Yuhe Nie, Kede Ma, et al. Come together, but not right now: A progressive strategy to boost low-rank adaptation. In *42nd International Conference on Machine Learning, ICML 2025*, 2025.

A Datasets

We evaluate three categories of tasks: commonsense reasoning, dialogue generation, and mathematical reasoning. The commonsense reasoning benchmarks include:

- **BoolQ** [5] is a reading-comprehension dataset of naturally occurring yes/no questions paired with Wikipedia passages, designed to evaluate semantic understanding and reasoning over text.
- **PIQA** (Physical Interaction Question Answering) [4] is a commonsense reasoning benchmark focused on physical knowledge, where the model must choose the more plausible solution to a goal-oriented question involving everyday object interactions and real-world physics.
- **SIQA** (Social Interaction Question Answering) [39] evaluates a model's understanding of social interactions and human intentions by requiring it to select the most plausible explanation or outcome in everyday situations.
- **ARC-c** (AI2 Reasoning Challenge, Challenge Set) [6] is a multiple-choice science benchmark composed of grade-school questions that require nontrivial reasoning and external knowledge.
- **ARC-e** (AI2 Reasoning Challenge, Easy Set) [6] is the easier subset of ARC, containing questions that can usually be answered with simpler reasoning or direct factual knowledge.
- **OBQA** [34] is a multiple-choice science benchmark in which each question requires combining a small set of "open-book" facts with additional commonsense reasoning.
- **HellaSwag** [47] tests whether a model can predict plausible next events in everyday situations by selecting the most realistic continuation of a given context.
- **WinoG** (WinoGrande) [1] is a large-scale commonsense reasoning dataset based on Winograd-style pronoun resolution and is designed to test whether a model can resolve ambiguous references using contextual and commonsense cues.

Dataset statistics for the commonsense reasoning benchmarks are summarized in Table 8.

For dialogue generation, we use **ConvAI2** [10], a persona-based open-domain conversation dataset in which the model must generate or select responses that are coherent, engaging, and consistent with the given persona. For mathematical reasoning, we use **GSM8K** [8]. Following [18], we train on MetaMath and evaluate on GSM8K [7].

B Baselines

We compare SMoA with a representative set of PEFT methods.

- **Prompt Tuning** [22] adapts pretrained language models by learning a small set of trainable prompt embeddings while freezing the backbone parameters.
- **P-Tuning v2** [28] injects trainable continuous prompts into every layer of a pretrained language model, yielding stronger parameter-efficient adaptation, especially at larger model scales.
- **LoRA** [16] inserts low-rank trainable matrices into existing linear layers, achieving strong performance while updating only a small number of parameters.
- **DoRA** [27] extends LoRA by decomposing the update into magnitude and direction components, allowing more expressive adaptation with a low parameter count.
- **MoRA** [19] replaces the standard LoRA structure with a higher-rank parameterization to improve expressive capacity under a constrained budget.
- **SSMLoRA** [45] augments LoRA with state-space-model-based structure to better capture long-range dependencies while keeping the number of trainable parameters small.
- **MeLoRA** [37] uses multiple small low-rank adapters to reduce memory cost while improving adaptation performance and robustness.
- **HiRA** [18] extends LoRA through hierarchical low-rank adaptation, enabling the model to capture both coarse- and fine-grained task-specific information.

C Evaluation Metric

We evaluate each task family with its standard automatic metrics. For BoolQ, PIQA, SIQA, ARC-c, ARC-e, OBQA, HellaSwag, and WinoGrande, we report test accuracy. For each example, the model generates a free-form answer, which we map to the task label set using task-specific answer keywords (e.g., "true" or "false" for BoolQ, or the corresponding option strings for multiple-choice tasks). The first valid matched keyword is taken as the prediction; if no valid keyword is found, the prediction is counted as incorrect. This extraction protocol follows prior work and yields a unified evaluation pipeline across all eight commonsense reasoning benchmarks [17, 27]. The "Average" score in Table 1 is the unweighted mean of the eight task accuracies.

For the conversational benchmark CONVAI2, we report BLEU [36], METEOR, ROUGE-L, and BERTScore [50]. In Table 2, BERT-P, BERT-R, and BERT-F1 denote BERTScore precision, recall, and F1, respectively. When we refer to the average conversational score, we mean the unweighted mean over the reported automatic metrics in Table 2.

For mathematical reasoning on GSM8K, we report final-answer accuracy, i.e., exact-match accuracy after extracting the numeric answer from the generated response using the standard answer format. A prediction is marked correct only when the extracted final answer exactly matches the ground-truth solution. Higher is better for all reported metrics.

D Implementation Details

All experiments are conducted on H800-80GB GPUs. We follow the training setup of [18] except for learning-rate adjustments. We implement SMoA on Llama-2-7B and Llama-3-8B using reference LoRA ranks $r = 16$ and $r = 32$. Unless otherwise stated, we use $K = 2$ aligned spectral blocks, so each block uses local rank $\rho = r/K$. For every adapted layer, the SVD of the frozen pretrained weight is computed once before training to determine the fixed spectrum-aware reordering and the corresponding block partition. This preprocessing step does not introduce additional trainable parameters and is not part of the inference path, although it does add a one-time offline cost before fine-tuning. For a layer $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ with $q = \min(d_{\text{out}}, d_{\text{in}})$, the preprocessing cost is the cost of one truncated/full SVD of that matrix, i.e., on the order of $O(d_{\text{out}} d_{\text{in}} q)$. During training, the total adapter parameter count becomes $K\big(\frac{r}{K}\frac{d_{\text{in}}}{K} + \frac{r}{K}\frac{d_{\text{out}}}{K}\big) = \frac{r}{K}(d_{\text{in}} + d_{\text{out}})$, which is a factor of K smaller than a rank-r LoRA layer. At inference time, the learned update can be mapped back to the original coordinates and merged into the base weight, so SMoA does not require a separate multi-branch execution path. We use the AdamW optimizer [31] with a learning rate of 0.001 and a warmup of 1000 steps. For BoolQ, PIQA, SIQA, ARC-c, ARC-e, OBQA, HellaSwag, and WinoGrande, we fine-tune for 5 epochs and evaluate every 100 steps, selecting the best validation checkpoint. For LoRA, DoRA, MoRA, HiRA, SSMLoRA, MeLoRA, and SMoA, we insert trainable updates into the query, key, value, and projection layers. Because SMoA uses fewer trainable parameters than rank-r LoRA in the current setting, the comparison should be interpreted as a sub-budget comparison rather than a matched-parameter comparison. All main results use the same evaluation pipeline; because they are single-run results, small gaps should be interpreted cautiously. We report the average of 5 random seeds (7,42,123,1234,12345) for SMoA. The hyperparameter configuration of SMoA is shown in Table 6 (Appendix H). The general hyperparameter configurations are shown in Table 7 (Appendix H).

E Additional Method Details

E.1 Spectrum-Aware Preprocessing Details

The spectrum-aware preprocessing step uses the frozen singular structure of W_0 only to define the permutations π_{out} and π_{in}. These permutations reorder output and input coordinates so that coordinates assigned similar spectral roles become contiguous in the reordered matrix $\widetilde{W}_0 = P_{\text{out}} W_0 P_{\text{in}}^{\top}$. In the equal-size setting with K blocks, the reordered row and column intervals $\mathcal{O}_k$ and $\mathcal{I}_k$ are the k-th contiguous intervals of lengths d_{out}/K and d_{in}/K, respectively. The corresponding frozen block anchors are then defined by

$$M_k = \widetilde{W}_0[\mathcal{O}_k, \mathcal{I}_k] \in \mathbb{R}^{\frac{d_{\text{out}}}{K} \times \frac{d_{\text{in}}}{K}}, \qquad k = 1, \ldots, K. \tag{10}$$

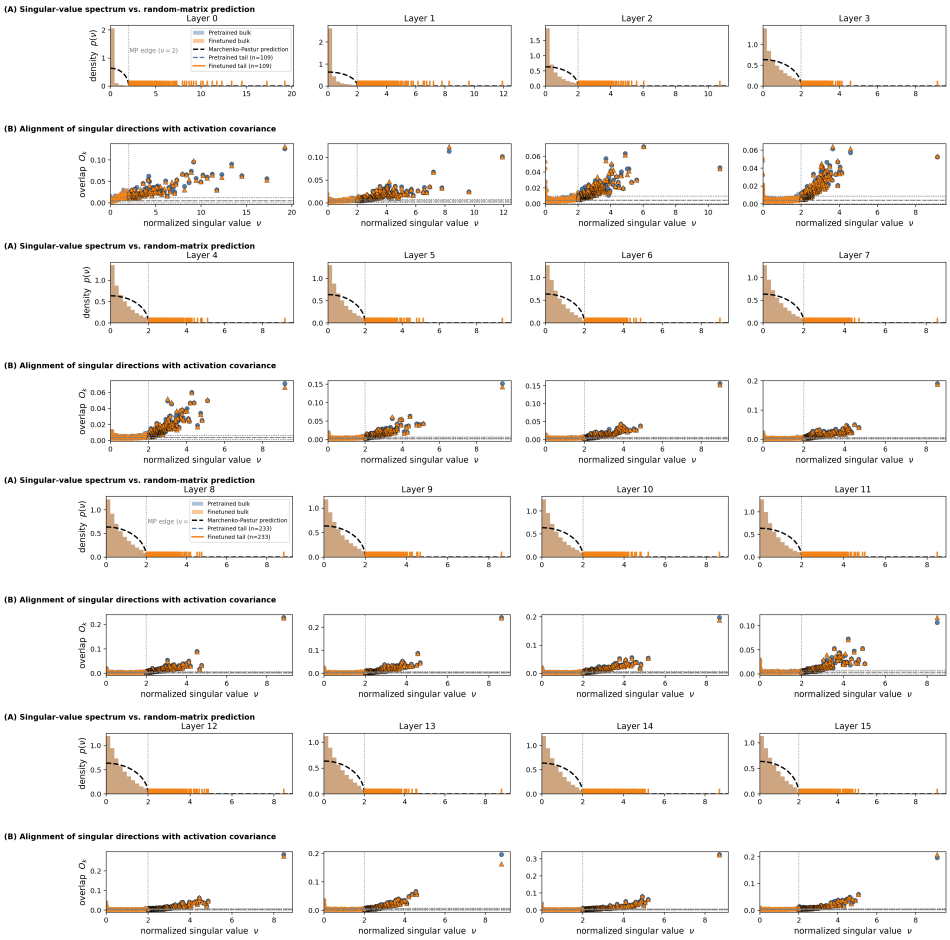

Figure 5: **Original Llama2-7B vs. finetuned spectra and overlaps for key weight matrix** (layer 0 to 15). Top: empirical distributions of normalized singular values ν; light brown denotes the pretrained model and orange denotes the finetuned model. The dashed curve is the theoretical random-matrix bulk. Bottom: the corresponding maximum overlap O_k between right singular vectors and activation-covariance eigenvectors, with circles for the pretrained model and triangles for the finetuned model. Gray lines indicate the bulk mean and $\pm 3\sigma$ intervals. The figure highlights how finetuning changes the right-tail spectrum and the alignment of tail singular directions with task-induced activation structure.

These anchors remain fixed during fine-tuning; only the low-rank factors attached to them are trained.

F More Results

F.1 Tail Spectrum Visualization

Layer-wise spectral changes before and after fine-tuning. Figure 5 and Figure 6 compare the layer-wise spectral distributions of the pretrained model and the finetuned SMoA model. Overall, we observe a consistent and highly structured spectral shift after fine-tuning. In the pretrained model, the singular spectrum is broadly distributed near the low-frequency/small-ν region, shown by the light-brown density. After SMoA fine-tuning, the spectrum becomes more concentrated and shifts toward a higher-ν band, shown by the orange density. This indicates that SMoA does not uniformly perturb all spectral directions; instead, it selectively amplifies a subset of singular directions that are weak or less dominant in the pretrained model.

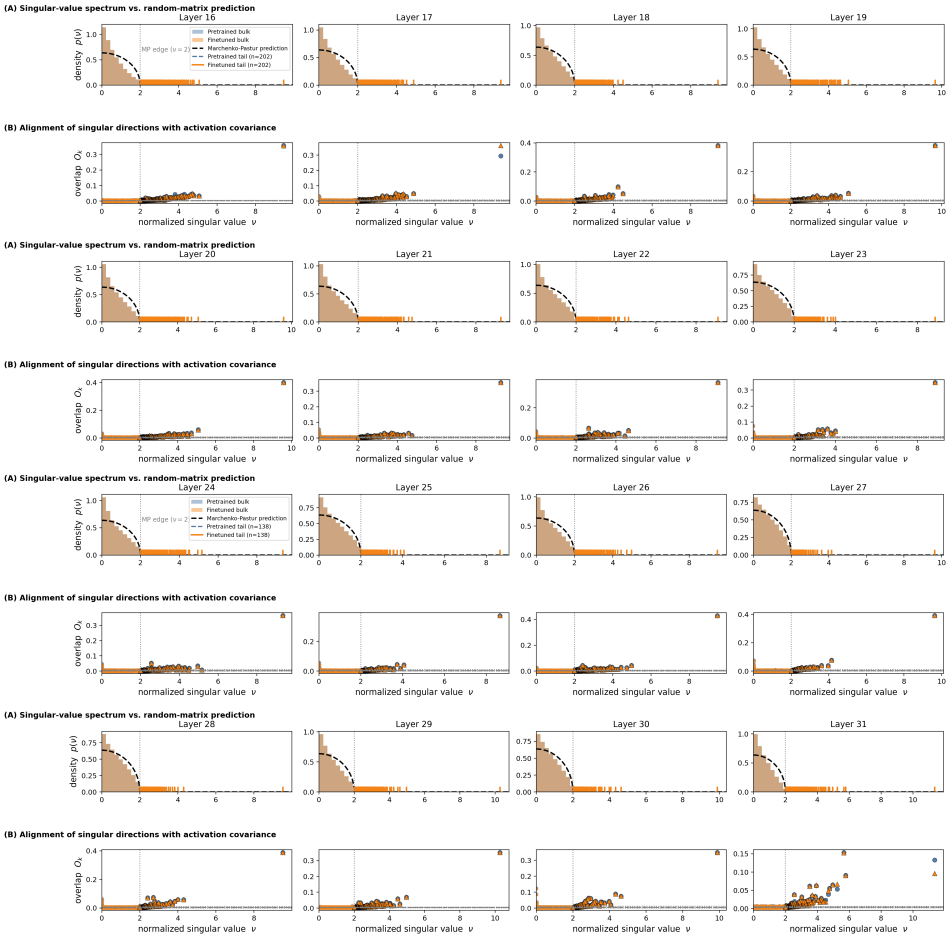

Figure 6: **Pretrained vs. finetuned spectra and overlaps on Llama2-7B key weight matrix** (layer 16 to 31). Top: empirical distributions of normalized singular values ν; light brown denotes the pretrained model and orange denotes the finetuned model. The dashed curve is the theoretical random-matrix bulk. Bottom: the corresponding maximum overlap O_k between right singular vectors and activation-covariance eigenvectors, with circles for the pretrained model and triangles for the finetuned model. Gray lines indicate the bulk mean and $\pm 3\sigma$ intervals. The figure highlights how finetuning changes the right-tail spectrum and the alignment of tail singular directions with task-induced activation structure.

At the earliest layers, i.e., Layer 0–3, the spectral change is relatively irregular compared with deeper layers. Layer 0 exhibits only a mild orange response, suggesting that the bottom layer is less affected by task-specific adaptation. From Layer 1 to Layer 3, the finetuned spectrum begins to form a visible bump around intermediate ν, while the pretrained spectrum still concentrates near the origin. The corresponding O_k values increase mainly in the intermediate-to-large ν region, implying that the learned updates start to align with non-dominant spectral components rather than the principal pretrained directions.

For the shallow-to-middle layers, Layer 4–11, the spectral shift becomes much clearer. The orange density consistently forms a sharp peak around $\nu \approx 2$, followed by a long but sparse tail. In contrast, the pretrained spectrum remains concentrated in the low-ν bulk. This separation suggests that SMoA introduces task-specific energy into directions that are not strongly expressed in the pretrained weights. The overlap plots further support this observation: O_k is close to zero for most low-ν components but grows noticeably in the intermediate and tail regions. In particular, Layers 6–11 show stronger isolated high-ν responses, indicating that these layers contain directions highly sensitive to fine-tuning.

In the middle layers, Layer 12–19, the phenomenon becomes even more pronounced. The orange spectrum is highly concentrated around a narrow band, whereas the pretrained density decays smoothly from the origin. This implies that fine-tuning reorganizes the spectral mass into a more compact set of directions. The O_k curves show that the dominant changes are not located in the pretrained spectral bulk, but rather in the right-side tail. Several layers, such as Layers 14, 17, 18, and 19, exhibit large O_k outliers at high ν, suggesting that SMoA learns a small number of highly influential spectral directions. These directions are likely associated with task-discriminative features that are underrepresented in the pretrained model.

For the deeper layers, Layer 20–27, the orange density becomes sharper and more peaked, while the pretrained spectrum keeps a relatively smooth low-ν distribution. This indicates that the effect of SMoA fine-tuning is increasingly selective in deeper transformer blocks. The overlap scores in these layers are mostly small in the bulk region but display clear spikes in the spectral tail. Such behavior is consistent with the intuition that later layers encode more task-specific abstractions: fine-tuning does not need to rewrite the entire representation space, but instead adjusts a sparse set of high-impact directions.

Finally, in the top layers, Layer 28–31, we observe the strongest spectral localization. The finetuned spectrum is sharply centered around a narrow ν interval, and the pretrained spectrum remains separated from it. The corresponding O_k values show clear activation in the intermediate-to-large ν region, especially in Layer 31, where several pronounced overlap responses appear. This suggests that the last layers are most directly responsible for adapting the pretrained representation to the downstream task. Importantly, the fine-tuning signal is not uniformly distributed across all singular components; rather, it is concentrated in a small number of tail or near-tail directions.

Taken together, these layer-wise observations provide empirical evidence that SMoA fine-tuning mainly modifies the tail and intermediate spectral components of the pretrained model, while leaving the dominant low-ν bulk largely intact. This supports the view that parameter-efficient fine-tuning can achieve adaptation by exploiting underutilized spectral directions instead of overwriting the principal pretrained subspace. The consistent emergence of orange peaks across layers further suggests that SMoA learns a structured spectral reallocation, concentrating task-specific information into a compact set of singular directions.

F.2 Spectral Structure

Figure 7 shows the singular-value histograms of the attention key weight matrix of Llama-2-7B based on SMoA fine-tuning in BoolQ (Similar results can be observed in the Query and Value parameter matrices). Rather than exhibiting a sharply low-rank spectrum, the weights display a clear separation between a bulk component and a set of outlying singular values. The bulk can be interpreted as a high-dimensional noise-like background, while the outliers correspond to structured directions learned during pretraining. Importantly, these outliers are not limited to only the top few singular modes; many layers contain a long tail of moderately large singular values that remain separated from the bulk. This suggests that the key projection matrices encode information in a distributed high-rank subspace, rather than concentrating all useful structure in a small number of dominant directions. Such a spectral pattern provides evidence that fine-tuning may require modifying not only the leading singular components, but also weaker tail directions that capture more fine-grained or task-specific features.

G Additional Proofs

Lemma 1 (Spectral Informativeness of Weights)**.** *Let $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ be a pretrained Transformer weight matrix with singular values*

$$\sigma_1(W_0) \geq \sigma_2(W_0) \geq \cdots \geq \sigma_m(W_0) \geq 0, \qquad m = \min(d_{\text{out}}, d_{\text{in}}). \tag{11}$$

Then prior empirical evidence suggests that W exhibits a long-tailed spectrum and is near full-rank in a numerical sense [40], in that

$$\text{rank}_\varepsilon(W_0) = \#\{i : \sigma_i(W_0) > \varepsilon\} \approx m, \tag{12}$$

while for any $r < m$,

$$\sum_{i=r+1}^{m} \sigma_i^2(W_0) \not\approx 0. \tag{13}$$

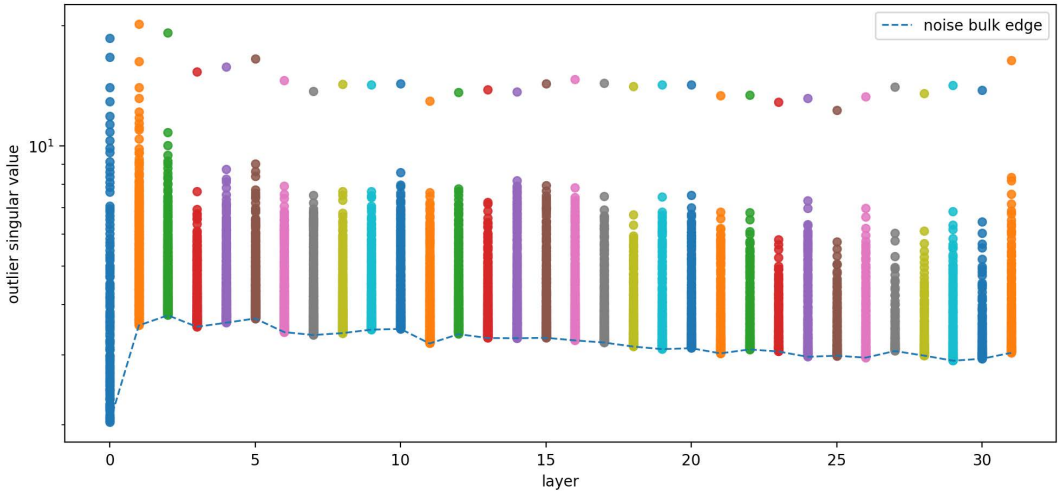

Figure 7: Attention key singular-value histograms of the Llama-2-7B based on SMoA fine-tuning in BoolQ across layers. The spectra exhibit a clear separation between the noise-like bulk and multiple outlier singular values. The persistent outliers, especially those in the spectral tail, indicate that the key projection weights contain rich high-rank structure.

G.1 Proof of Lemma 1

Proof. Write the singular value decomposition of W as

$$W_0 = U\Sigma V^\top, \qquad \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_m). \tag{14}$$

By the empirical observations of Staats et al. [40], the singular-value spectra of pretrained Transformer weight matrices exhibit a long-tailed profile rather than a sharp low-rank collapse. Equivalently, there is no rapid transition of the form

$$\sigma_1 \geq \cdots \geq \sigma_r \gg \sigma_{r+1} \approx \cdots \approx \sigma_m \approx 0 \tag{15}$$

for some small $r \ll m$; instead, a substantial portion of the spectrum remains above numerical noise level. Hence

$$\mathrm{rank}_\varepsilon(W_0) = \#\{i : \sigma_i(W_0) > \varepsilon\} \approx m, \tag{16}$$

which gives the claimed near-full-rank statement in the numerical sense.

Next, for any $r < m$, the Eckart–Young–Mirsky theorem gives

$$\|W_0 - W_0^{(r)}\|_F^2 = \sum_{i=r+1}^{m} \sigma_i^2(W_0) = E_{\mathrm{tail}}(r). \tag{17}$$

The same empirical study further shows that removing small singular values can increase perplexity, especially after alignment or downstream fine-tuning. Therefore, the lower-tail singular directions cannot be identified with pure noise in general, which implies that $E_{\mathrm{tail}}(r)$ should not be assumed negligible a priori.

Combining the long-tailed spectrum,

$$\mathrm{rank}_\varepsilon(W_0) \approx m, \tag{18}$$

with the non-negligible effect of truncating

$$\sum_{i=r+1}^{m} \sigma_i^2(W_0), \tag{19}$$

we conclude that pretrained Transformer weight matrices are more faithfully described as long-tailed and numerically near full-rank.

□

G.2 Proof of Proposition 1

Proof. For each block, the local update is

$$\Delta M_k = (B_k A_k) \odot M_k. \tag{20}$$

By the standard rank inequality for Hadamard products,

$$\operatorname{rank}(\Delta M_k) = \operatorname{rank}\big((B_k A_k) \odot M_k\big) \le \operatorname{rank}(B_k A_k)\ \operatorname{rank}(M_k). \tag{21}$$

Since

$$B_k A_k \in \mathbb{R}^{\frac{d_{\text{out}}}{K} \times \frac{d_{\text{in}}}{K}} \tag{22}$$

is formed from factors of inner dimension ρ, we have

$$\operatorname{rank}(B_k A_k) \le \rho. \tag{23}$$

Therefore,

$$\operatorname{rank}(\Delta M_k) \le \rho\ \operatorname{rank}(M_k). \tag{24}$$

On the other hand, since

$$\Delta M_k \in \mathbb{R}^{\frac{d_{\text{out}}}{K} \times \frac{d_{\text{in}}}{K}}, \tag{25}$$

its rank is also bounded by the block size:

$$\operatorname{rank}(\Delta M_k) \le s_k, \qquad s_k = \min\left(\frac{d_{\text{out}}}{K}, \frac{d_{\text{in}}}{K}\right). \tag{26}$$

Combining the two bounds yields

$$\operatorname{rank}(\Delta M_k) \le \min(s_k,\ \rho\ \operatorname{rank}(M_k)). \tag{27}$$

Now, by construction,

$$\Delta W = P_{\text{out}}^{\top}\, \widetilde{\Delta W}\, P_{\text{in}}, \qquad \widetilde{\Delta W} = \operatorname{blkdiag}(\Delta M_1, \ldots, \Delta M_K). \tag{28}$$

Since permutation matrices preserve rank and rank is additive over block-diagonal matrices,

$$\operatorname{rank}(\Delta W) = \operatorname{rank}(\widetilde{\Delta W}) = \sum_{k=1}^{K} \operatorname{rank}(\Delta M_k) \le \sum_{k=1}^{K} \min(s_k,\ \rho\ \operatorname{rank}(M_k)). \tag{29}$$

Define

$$U := \sum_{k=1}^{K} \min(s_k,\ \rho\ \operatorname{rank}(M_k)). \tag{30}$$

Then clearly

$$U \ge 0. \tag{31}$$

Moreover, using $\min(a, b) \le a$ for each block,

$$U \le \sum_{k=1}^{K} s_k. \tag{32}$$

Under the equal-size partition,

$$\sum_{k=1}^{K} s_k = K\ \min\left(\frac{d_{\text{out}}}{K}, \frac{d_{\text{in}}}{K}\right) = \min(d_{\text{out}}, d_{\text{in}}). \tag{33}$$

Hence

$$\operatorname{rank}(\Delta W) \le U \le \min(d_{\text{out}}, d_{\text{in}}). \tag{34}$$

For standard LoRA, the update takes the form

$$\Delta W_{\text{LoRA}} = BA, \qquad B \in \mathbb{R}^{d_{\text{out}} \times r}, \quad A \in \mathbb{R}^{r \times d_{\text{in}}}, \tag{35}$$

and therefore

$$\operatorname{rank}(\Delta W_{\text{LoRA}}) = \operatorname{rank}(BA) \le r. \tag{36}$$

Thus, whenever

$$U > r, \tag{37}$$

the analytic rank ceiling of SMoA is strictly larger than that of rank-r LoRA. □

G.3 Proof of Corollary 1

Corollary 1 (Rank-Ceiling Separation under Full-Rank Local Anchors)**.** *Assume the equal-size partition setting and define*

$$s = \min\left(\frac{d_{out}}{K}, \frac{d_{in}}{K}\right). \tag{38}$$

If each reordered anchor block is full-rank within its local dimensions, i.e.,

$$\operatorname{rank}(M_k) = s, \qquad k = 1, \dots, K, \tag{39}$$

then the SMoA analytic rank upper bound in Equation (8) *reduces to*

$$U_{\mathrm{nf}} := \sum_{k=1}^{K} \min\Big(s, \frac{r}{K}\operatorname{rank}(M_k)\Big) = K \min\Big(s, \frac{r}{K}s\Big). \tag{40}$$

Moreover, if

$$K < \min(d_{out}, d_{in}) \qquad \textit{and} \qquad r < \min(d_{out}, d_{in}), \tag{41}$$

then

$$U_{\mathrm{nf}} > r. \tag{42}$$

Therefore, under full-rank local anchors, SMoA admits a strictly larger analytic rank ceiling than rank-r LoRA.

Proof. Under the assumption

$$\operatorname{rank}(M_k) = s, \qquad k = 1, \dots, K, \tag{43}$$

Equation (8) becomes

$$U_{\mathrm{nf}} = \sum_{k=1}^{K} \min\Big(s, \frac{r}{K}s\Big) = K \min\Big(s, \frac{r}{K}s\Big). \tag{44}$$

We consider two cases.

Case 1: $r < K$. Then $\frac{r}{K} < 1$, so

$$U_{\mathrm{nf}} = K \cdot \frac{r}{K}s = rs. \tag{45}$$

Since

$$K < \min(d_{\mathrm{out}}, d_{\mathrm{in}}), \tag{46}$$

we have

$$s = \min\left(\frac{d_{\mathrm{out}}}{K}, \frac{d_{\mathrm{in}}}{K}\right) > 1, \tag{47}$$

which implies

$$U_{\mathrm{nf}} = rs > r. \tag{48}$$

Case 2: $r \geq K$. Then $\frac{r}{K} \geq 1$, hence

$$U_{\mathrm{nf}} = Ks = \min(d_{\mathrm{out}}, d_{\mathrm{in}}). \tag{49}$$

Using the assumption

$$r < \min(d_{\mathrm{out}}, d_{\mathrm{in}}), \tag{50}$$

we obtain

$$U_{\mathrm{nf}} > r. \tag{51}$$

Therefore, in both cases,

$$U_{\mathrm{nf}} > r. \tag{52}$$

Since the LoRA analytic rank ceiling is r, SMoA admits a strictly larger analytic rank upper bound under the stated full-rank block condition. □

G.4 Proof of Proposition 2

Proposition 2 (Expressive Family Separation)**.** *For a reference rank budget r, define the standard LoRA family*

$$\mathcal{F}_{\mathrm{LoRA}}(r) = \left\{ BA : A \in \mathbb{R}^{r \times d_{in}},\ B \in \mathbb{R}^{d_{out} \times r} \right\}. \tag{53}$$

Define also the SMoA family induced by the fixed preprocessing $(P_{out}, P_{in}, \{M_k\}_{k=1}^{K})$ as

$$\begin{aligned} \mathcal{F}_{\mathrm{SMoA}}(W_0; r, K) = \Big\{ P_{out}^{\top} \operatorname{blkdiag}\big((B_1 A_1) \odot M_1, \dots, (B_K A_K) \odot M_K\big) P_{in} : \\ A_k \in \mathbb{R}^{\rho \times d_{in}/K},\ B_k \in \mathbb{R}^{d_{out}/K \times \rho},\ k = 1, \dots, K \Big\}, \end{aligned} \tag{54}$$

where $\rho = r/K$. Now let

$$C_k \in \mathbb{R}^{\frac{d_{out}}{K} \times \frac{d_{in}}{K}}, \qquad \operatorname{rank}(C_k) \le \rho, \qquad k = 1, \dots, K, \tag{55}$$

and define the block-aligned target update in reordered coordinates by

$$\widetilde{\Delta W}^{\star} = \operatorname{blkdiag}(C_1 \odot M_1, \dots, C_K \odot M_K). \tag{56}$$

Then:

1. $$P_{out}^{\top} \widetilde{\Delta W}^{\star} P_{in} \in \mathcal{F}_{\mathrm{SMoA}}(W_0; r, K). \tag{57}$$

2. *If*
$$\operatorname{rank}(\widetilde{\Delta W}^{\star}) > r, \tag{58}$$
then
$$P_{out}^{\top} \widetilde{\Delta W}^{\star} P_{in} \notin \mathcal{F}_{\mathrm{LoRA}}(r). \tag{59}$$

Therefore, any block-aligned anchor-modulated target whose rank exceeds r serves as a witness that

$$\mathcal{F}_{\mathrm{SMoA}}(W_0; r, K) \setminus \mathcal{F}_{\mathrm{LoRA}}(r) \neq \varnothing. \tag{60}$$

Proof. Since

$$\operatorname{rank}(C_k) \le \rho, \tag{61}$$

each matrix C_k admits a rank-ρ factorization

$$C_k = B_k A_k, \qquad A_k \in \mathbb{R}^{\rho \times d_{\mathrm{in}}/K}, \quad B_k \in \mathbb{R}^{d_{\mathrm{out}}/K \times \rho}. \tag{62}$$

Substituting these factorizations into the SMoA construction gives

$$\begin{aligned} &P_{\mathrm{out}}^{\top} \operatorname{blkdiag}\big((B_1 A_1) \odot M_1, \dots, (B_K A_K) \odot M_K\big) P_{\mathrm{in}} \\ &\quad = P_{\mathrm{out}}^{\top} \operatorname{blkdiag}(C_1 \odot M_1, \dots, C_K \odot M_K) P_{\mathrm{in}} = P_{\mathrm{out}}^{\top} \widetilde{\Delta W}^{\star} P_{\mathrm{in}}. \end{aligned} \tag{63}$$

Hence

$$P_{\mathrm{out}}^{\top} \widetilde{\Delta W}^{\star} P_{\mathrm{in}} \in \mathcal{F}_{\mathrm{SMoA}}(W_0; r, K). \tag{64}$$

Next, every matrix in $\mathcal{F}_{\mathrm{LoRA}}(r)$ has rank at most r, because for any

$$\Delta W_{\mathrm{LoRA}} = BA \quad \text{with} \quad A \in \mathbb{R}^{r \times d_{\mathrm{in}}},\ B \in \mathbb{R}^{d_{\mathrm{out}} \times r}, \tag{65}$$

we have

$$\operatorname{rank}(\Delta W_{\mathrm{LoRA}}) = \operatorname{rank}(BA) \le r. \tag{66}$$

Moreover, permutation matrices preserve rank, so

$$\operatorname{rank}\Big(P_{\mathrm{out}}^{\top} \widetilde{\Delta W}^{\star} P_{\mathrm{in}}\Big) = \operatorname{rank}(\widetilde{\Delta W}^{\star}). \tag{67}$$

Therefore, if

$$\operatorname{rank}(\widetilde{\Delta W}^{\star}) > r, \tag{68}$$

then

$$\operatorname{rank}\Big(P_{\mathrm{out}}^{\top} \widetilde{\Delta W}^{\star} P_{\mathrm{in}}\Big) > r, \tag{69}$$

which rules out membership in $\mathcal{F}_{\mathrm{LoRA}}(r)$. Thus

$$P_{\mathrm{out}}^{\top} \widetilde{\Delta W}^{\star} P_{\mathrm{in}} \notin \mathcal{F}_{\mathrm{LoRA}}(r). \tag{70}$$

This proves the claim.

□

G.5 Proof of Corollary 2

Corollary 2 (Spectral Approximation Gap)**.** *Let*

$$\widetilde{\Delta W}^{\star} = \mathrm{blkdiag}(C_1 \odot M_1, \ldots, C_K \odot M_K) \tag{71}$$

be any block-aligned target from Proposition 2, and let

$$\sigma_1(\widetilde{\Delta W}^{\star}) \geq \sigma_2(\widetilde{\Delta W}^{\star}) \geq \cdots \tag{72}$$

denote its singular values. If

$$\mathrm{rank}(\widetilde{\Delta W}^{\star}) > r, \tag{73}$$

then

$$\inf_{\Delta \in \mathcal{F}_{\mathrm{LoRA}}(r)} \left\| \Delta - P_{out}^{\top} \widetilde{\Delta W}^{\star} P_{in} \right\|_F^2 = \sum_{j>r} \sigma_j(\widetilde{\Delta W}^{\star})^2 > 0, \tag{74}$$

whereas

$$\inf_{\Delta \in \mathcal{F}_{\mathrm{SMoA}}(W_0;r,K)} \left\| \Delta - P_{out}^{\top} \widetilde{\Delta W}^{\star} P_{in} \right\|_F^2 = 0. \tag{75}$$

Therefore, rank-r LoRA incurs an irreducible spectral-tail approximation error on this witness family, while SMoA realizes the target exactly.

Proof. By Proposition 2,

$$P_{\mathrm{out}}^{\top} \widetilde{\Delta W}^{\star} P_{\mathrm{in}} \in \mathcal{F}_{\mathrm{SMoA}}(W_0; r, K), \tag{76}$$

and therefore

$$\inf_{\Delta \in \mathcal{F}_{\mathrm{SMoA}}(W_0;r,K)} \left\| \Delta - P_{\mathrm{out}}^{\top} \widetilde{\Delta W}^{\star} P_{\mathrm{in}} \right\|_F^2 = 0. \tag{77}$$

For LoRA, note that

$$\mathcal{F}_{\mathrm{LoRA}}(r) = \{X \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}} : \mathrm{rank}(X) \leq r\}. \tag{78}$$

Moreover, permutation matrices preserve both the Frobenius norm and the singular values. Hence

$$\begin{aligned} &\inf_{\Delta \in \mathcal{F}_{\mathrm{LoRA}}(r)} \left\| \Delta - P_{\mathrm{out}}^{\top} \widetilde{\Delta W}^{\star} P_{\mathrm{in}} \right\|_F^2 \\ &\quad = \inf_{\mathrm{rank}(X) \leq r} \left\| X - \widetilde{\Delta W}^{\star} \right\|_F^2. \end{aligned} \tag{79}$$

By the Eckart–Young–Mirsky theorem, the right-hand side is exactly

$$\sum_{j>r} \sigma_j(\widetilde{\Delta W}^{\star})^2. \tag{80}$$

Since

$$\mathrm{rank}(\widetilde{\Delta W}^{\star}) > r, \tag{81}$$

at least one singular value beyond the first r is nonzero, and thus

$$\sum_{j>r} \sigma_j(\widetilde{\Delta W}^{\star})^2 > 0. \tag{82}$$

This proves the result.

□

H Training Hyperparameters

I Limitations

SMoA requires a one-time spectral preprocessing step for each adapted pretrained weight matrix. This preprocessing does not add trainable parameters or inference-time latency after merging, but it introduces additional offline computation and storage for the fixed permutations and block anchors. The cost may become non-negligible for very large models or for settings where many checkpoints must be adapted independently.

Table 6: Hyperparameters for SMoA.

Hyperparameter	Value
Optimizer	AdamW
Weight Decay	0
Base Model	[Llama-2-7B, Llama-3-8B]
Learning Rate	[0.0001, 0.0002]
r	[2,4,8,16,24,32]
Warm Up	1000 steps
Batch Size	32
Target Modules	q_proj, k_proj, v_proj, up_proj, down_proj
Evaluation Steps	100

Table 7: General hyperparameters used across all experiments.

Hyperparameter	Value
Optimizer	AdamW
Weight Decay	0
Base Model	Llama-3-8B
Warm Up	1000 steps
Target Modules	q_proj, k_proj, v_proj, up_proj, down_proj
Evaluation Steps	100

Table 8: The detailed statistics of commonsense reasoning datasets.

Dataset	Data Number	Type
Train	170,300	Mixed
Validation	120	Mixed
Test		
BoolQ	3,270	Yes/No
PIQA	1,830	Option
SIQA	1,954	Option
HellaSwag	10,042	Option
WinoGrande	1,267	Option
ARC-e	2,376	Option
ARC-c	1,172	Option
OBQA	500	Option

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Section 3 introduces SMoA and its spectrum-aware block construction, Section 3.2 formalizes the parameter count, rank-capacity, and expressivity claims, and Section 4 reports results on different tasks.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses limitations in Appendix I.

   Guidelines:

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

tant role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The main theoretical statements are stated in Section 3.2, and the full assumptions and proofs are provided in Appendix G.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 3 specifies the SMoA construction, and Appendices A, B, C, D, and H describe datasets, baselines, metrics, training setup, seeds, and hyperparameters needed to reproduce the main experimental protocol.

   Guidelines:

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

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: We provide an anonymous code repository containing the implementation, scripts, and usage instructions for reproducing the experimental results.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Appendix D specifies the adapted layers, optimizer, learning rate, warmup, epochs, validation-checkpoint selection, seeds for SMoA, and evaluation frequency; Appendix H provides the detailed hyperparameter tables.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Table 1 reports SMoA results as mean accuracy with standard deviation, and Appendix D states that SMoA is averaged over five random seeds.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: This paper reports the GPU type used for all experiments in the Appendix D.

   Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: We adhere to all the ethics guidelines.

Guidelines:

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

10. **Broader impacts**

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

Answer: [No]

Justification: The paper focuses on a PEFT algorithm and does not include a dedicated broader-impact discussion of potential positive and negative societal effects.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: This paper poses no such risks.

Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Existing datasets and code are properly cited.

    Guidelines:

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

13. **New assets**

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

    Answer: [N/A]

    Justification: Paper does not exist new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: This paper does not involve LLMs in any meaningful capacity.

    Guidelines:

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