FARSS: Fisher-Optimized Adaptive Low-Rank and Singular-Vector Selection for Knowledge-Preserving Fine-Tuning

Anonymous ACL submission

Abstract

Parameter-efficient fine-tuning (PEFT) has become a prevalent approach for adapting large language models (LLMs). However, low-rank adaptation methods face an inherent trade-off: improving target task performance can compromise pre-trained world knowledge, while aggressively constraining updates to preserve world knowledge may hinder improvements in the target task. Furthermore, most current methods fail to account for layer-wise differences in adaptation sensitivity, resulting in suboptimal preservation of world knowledge and task adaptation. To address these challenge, we propose **F**isher-Optimized **A**daptive Low-**R**ank and **S**ingular-Vector **S**election (FARSS), an effective framework for knowledge-preserving fine-tuning. This framework introduces two key innovations. First, we propose a Fisher-guided adaptive rank allocation strategy, which assigns smaller ranks to shallow layers that are critical for preserving world knowledge, and larger ranks to deep layers that are essential for task adaptation. Second, we introduce a task-aware initialization method that integrates singular value information with layer-specific second-order statistics estimated from activation and gradient covariances, enabling efficient and task-sensitive low-rank updates. We evaluated several models across various tasks, and the experimental results show that our approach outperforms existing PEFT methods, including LoRA, Corda, and KaSA, achieving a balance between preserving world knowledge and enhancing target task performance. The code is available at `https://anonymous.4open.science/r/FARSS-394B`.

1 Introduction

Large language models (LLMs) have demonstrated strong capabilities across a wide range of natural language processing tasks (Devlin et al., 2019; Radford et al., 2018). Adapting these models to target tasks, however, remains challenging due to their scale and the associated computational and memory costs (Ding et al., 2023b). To make the adaptation process more resource-efficient, parameter-efficient fine-tuning (PEFT) techniques have been introduced (Han et al., 2024). These methods significantly reduce the number of trainable parameters by only fine-tuning newly added adapters (Hu et al., 2022; Houlsby et al., 2019) or tokens (Lester et al., 2021; Li and Liang, 2021), while keeping the original pre-trained weights frozen.

Among PEFT methods, low-rank adaptation (LoRA) (Hu et al., 2022) is one of the most widely adopted approaches. LoRA is motivated by the observation that weight updates during fine-tuning often exhibit a low-rank structure, and it introduces low-rank matrices into linear layers to efficiently parameterize task-specific updates. Despite its effectiveness, LoRA does not explicitly consider the preservation of pre-trained world knowledge. In practice, adapting LLMs with LoRA can still lead to catastrophic forgetting (Ren et al., 2024), where improvements on target tasks are accompanied by degradation on knowledge-intensive or general-domain evaluations.

Recent work has explored how PEFT methods can better balance task adaptation and the preservation of world knowledge (Yang et al., 2024; Wang et al., 2024a). Despite these significant efforts, two limitations remain. First, most existing approaches apply adaptation in a largely uniform manner across layers (Yang et al., 2024), even though Transformer layers are known to play heterogeneous roles and to exhibit substantially different sensitivities to fine-tuning updates (Guo et al., 2025; Chen and et al., 2024). Second, the selection of low-rank update directions is commonly based on weight geometry, activation statistics, or spectral criteria (Meng et al., 2024). These criteria are not explicitly aligned with the fine-tuning objective. As a result, under a fixed low-rank budget, it remains unclear which layers should be adapted

and which update directions are most responsible for improving task performance without degrading world knowledge.

To address these issues, we propose **F**isher-Optimized **A**daptive Low-**R**ank and **S**ingular-Vector **S**election (FARSS). This PEFT framework explicitly considers both layer-wise heterogeneity and objective-aware direction selection. FARSS differentiates the adaptation behavior of shallow and deep layers. It applies more conservative updates to layers that are empirically more sensitive to disruptions in world knowledge, while allowing greater adaptation capacity in task-relevant layers. In addition, FARSS selects low-rank update directions based on their relevance to the fine-tuning objective. This design enables more effective use of a limited parameter budget. By jointly determining where to adapt and how to adapt, FARSS provides a structured approach to fine-tuning. This approach better balances downstream task performance with preservation of world knowledge.

In summary, our main contributions are summarized as follows:

- We highlight the critical role of shallow layers in preserving world knowledge in PEFT. Using Fisher statistics, we quantify layer sensitivity to ensure minimal disruption to pre-trained world knowledge.
- We introduce FARSS, a novel PEFT framework that dynamically allocates low-rank updates across layers. It prioritizes shallow layers for world knowledge preservation and deep layers for task adaptation. The task-aware initialization uses singular value decomposition and second-order statistics to optimize updates, balancing world knowledge preservation and task performance.
- Extensive experiments across multiple tasks and model families demonstrate that FARSS consistently improves the balance between target task performance and world knowledge preservation.

2 Preliminaries and Related Work

Preliminaries on Low-Rank Adaptation. PEFT adapts large language models with a small number of trainable parameters. LoRA (Hu et al., 2022) keeps the pre-trained weight $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ frozen and learns a low-rank update:

$$W' = W + \Delta W = W + BA, \qquad (1)$$

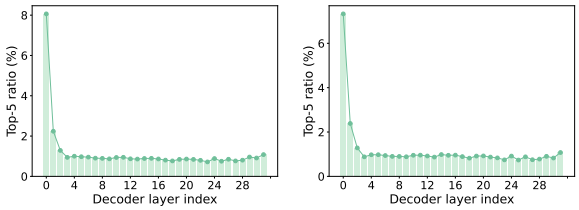
where $B \in \mathbb{R}^{d_{\text{out}} \times r}$ and $A \in \mathbb{R}^{r \times d_{\text{in}}}$ with $r \ll \min(d_{\text{out}}, d_{\text{in}})$. A common initialization sets A randomly and B to zeros so that $\Delta W = 0$ at the start of training.

While PEFT improves target-task performance, preserving pre-trained world knowledge and general capabilities remains challenging. In practice, even low-rank updates can induce substantial forgetting when adaptation is too aggressive (Kalajdzievski, 2024).

Parameter-Efficient Fine-Tuning. Since LLMs contain tens or even hundreds of billions of parameters (Naveed et al., 2025), full fine-tuning is computationally and memory intensive (Zhao et al., 2024), motivating PEFT methods that update only a small subset of parameters (Ding et al., 2023b; Xu et al., 2023). Among them, LoRA introduces trainable low-rank matrices into frozen weights for efficient adaptation with minimal overhead (Hu et al., 2022; Mahabadi et al., 2021). Subsequent LoRA based methods improve performance and efficiency along three main directions: initialization and subspace priors, including PiSSA (Meng et al., 2024), MiLoRA (Wang et al., 2025), and CorDA (Yang et al., 2024); adaptive resource and rank allocation, exemplified by AdaLoRA (Zhang et al., 2023b); and system level or architectural extensions, such as quantization or expert routing in QLoRA (Dettmers et al., 2023) and GOAT (Fan et al., 2025), as well as alternative parameterizations like VeRA (Kopiczko et al., 2023) and DoRA (Liu et al., 2024a).

Knowledge Preservation and Forgetting. Despite the efficiency of PEFT, fine-tuning may still degrade pre-trained linguistic or factual knowledge (Yang et al., 2024; Wang et al., 2024b). Prior work mitigates this by constraining updates or isolating task-specific capacity, for example, restricting LoRA updates via orthogonal projection to avoid interference with knowledge-bearing directions (Xiong and Xie, 2025), or routing task signals into dedicated LoRA experts while freezing the base model (Dou et al., 2024). These approaches show that how LoRA interacts with pre-trained representations is central to knowledge preservation, motivating structure-aware adaptation and initialization (Zhu et al., 2024).

Adaptive Rank Allocation in Low-rank Adapters. Static, layer-uniform ranks in LoRA overlook heterogeneous layer roles (Liu et al., 2024b; Hu et al., 2025). Adaptive-rank methods address this by reallocating capacity based on the im-



(a) q-proj (b) k-proj

Figure 1: Ratio of the top-5 singular value energy across decoder layers. shallow layers exhibit significantly higher energy concentration, indicating more stable and low-rank representations.

Strategy	World Knowledge			Math Benchmark	
	NQ-Open	TriviaQA	WebQS	GSM8K	Math
No freezing (origin)	14.99	52.51	5.86	—	—
Freeze shallow 1/3	13.05	50.82	7.68	31.16	4.94
Freeze deep 1/3	5.21	45.07	6.84	40.71	5.22

Table 1: Effect of freezing different layer segments during fine-tuning on **LLaMA2-7B**. **Freeze shallow 1/3** freezes the lowest third of decoder layers , while **freeze deep 1/3** freezes the highest third.

portance of layers or matrices. AdaLoRA (Zhang et al., 2023b) redistributes rank budgets by pruning less important singular components, while later work refines importance estimation with gradient-based or sensitivity-aware signals for more stable rankings (Refael et al., 2024; Valipour et al., 2023; Gu et al., 2025; He et al., 2025a). Other variants jointly optimize rank allocation and parameterization to match better task sensitivity (Ding et al., 2023a; Zhang et al., 2023a; Huang et al., 2025; He et al., 2025b).

3 Methodology

In this section, we present FARSS, as illustrated in Figure 2, a layer-aware fine-tuning framework that targets the trade-off between task adaptation and world knowledge preservation. FARSS implements two complementary strategies. It employs Fisher-based importance measures to adaptively allocate a fixed low-rank budget across shallow layers, thereby minimizing disruption to world knowledge during fine-tuning. Simultaneously, FARSS utilizes task-aware singular vector directions to initialize the low-rank adapter, enhancing task adaptability while preserving world knowledge.

3.1 The Importance of Shallow Layers for World Knowledge Preservation

Recent studies show that Transformer-based language models exhibit clear functional stratification across depth, with different layers encoding representations at distinct abstraction levels (Pan et al., 2024; Skean et al., 2025; Jin et al., 2025). As shown in Figure 1, shallow layers exhibit a higher concentration of dominant singular directions. This suggests that shallow representations contain more stable and broadly reused components that are closely tied to pretrained world knowledge (Zeng et al., 2025). Consequently, fine-tuning shallow layers without accounting for their stored world knowledge risks significant disruption to the pre-trained representations.

This observation is further validated through experiments with segment-wise freezing under math fine-tuning. As shown in Table 1, freezing shallow layers better preserves performance on world knowledge benchmarks, including TriviaQA (Joshi et al., 2017), NQ Open (Lee et al., 2019), and WebQS (Berant et al., 2013), but yields smaller gains on math benchmarks such as GSM8K (Cobbe et al., 2021) and Math (Yu et al., 2023). Therefore, the preservation of knowledge is critically dependent on shallow layers, which require more cautious updates.

The above analysis indicates that shallow layers are crucial for retaining world knowledge, while the deep layers are more adaptable to task changes. Under a fixed low-rank budget, a uniform allocation can lead to unnecessary updates in shallow layers and compromise world-knowledge preservation. Informed by prior work on depthwise functional stratification (Yao et al., 2024; Jin et al., 2024; Sun et al., 2025), FARSS derives its layer partition by optimizing a key trade-off. The search for the boundary is confined to the shallow one-eighth to one-quarter of layers, to establish an equilibrium that maximizes task adaptation while minimizing the loss of world knowledge.

3.2 Fisher Inverse Rank Allocation for Knowledge Preservation

Due to the uneven contribution of shallow layers to knowledge preservation, FARSS assigns lower ranks to more important layers based on Fisher estimates of layer importance for each task. This reverse allocation strategy in adaptive ranking maximizes the preservation of world knowledge.

Task-derived Fisher statistics. Within shallow layers $\mathcal{E}$, FARSS assigns less capacity to more sensitive weights to limit disruptive shallow layers updates under a fixed budget. FARSS quantifies the sensitivity of each shallow layer's weight to the

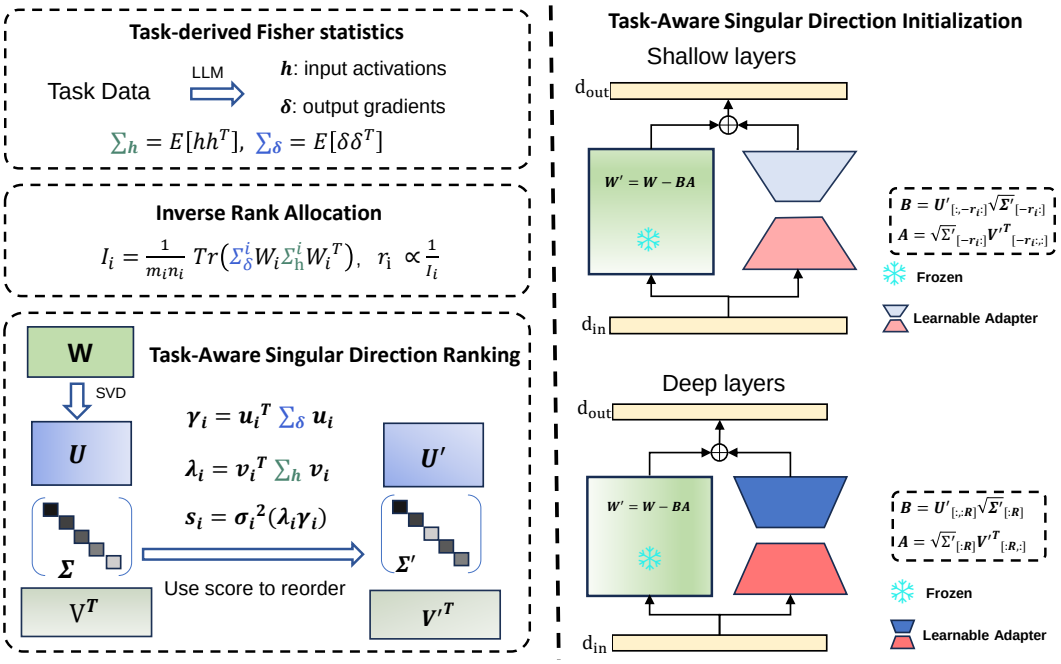


Figure 2: An overall illustration of FARSS. We apply singular value decomposition(SVD) to each target weight matrix and use task-derived covariance statistics to score and order its singular vectors (Left). Shallow layers initialize the learnable adapter with lower-scored singular vectors to preserve world knowledge (Top right). In contrast, deep layers use higher-scored singular vectors to facilitate fine-tuning adaptation (Bottom right). The shallow–deep partition is determined by a lightweight boundary search (Section 3.1).

target-task objective using Fisher information, and redistributes a fixed LoRA rank budget within $\mathcal{E}$. FARSS samples a small set of examples from the fine-tuning dataset and feeds them into the target LLM. Consider the i-th linear layer in the shallow decoder blocks with weight matrix $\mathbf{W}_i \in \mathbb{R}^{m_i \times n_i}$. Its forward mapping is $\mathbf{y}_i = \mathbf{W}_i \mathbf{h}_i$, where $\mathbf{h}_i$ denotes the input activation, and the backpropagated gradient is $\boldsymbol{\delta}_i = \partial \mathcal{L} / \partial \mathbf{y}_i$. Following K-FAC (Martens and Grosse, 2015), FARSS approximates the curvature of layer i with a Kronecker-factored Fisher approximation:

$$\mathbf{F}_i \approx \boldsymbol{\Sigma}_h^i \otimes \boldsymbol{\Sigma}_\delta^i, \tag{2}$$

where $\boldsymbol{\Sigma}_h^i = \mathbb{E}[\mathbf{h}_i \mathbf{h}_i^\top]$ and $\boldsymbol{\Sigma}_\delta^i = \mathbb{E}[\boldsymbol{\delta}_i \boldsymbol{\delta}_i^\top]$ denote the covariance matrices of activations and gradients, respectively. In practice, FARSS estimates these expectations by averaging over a few task mini-batches with frozen weights. A derivation is provided in Appendix A.1.1. With this Fisher approximation, we can characterize how small weight perturbations affect the loss.

For a small perturbation $\Delta \mathbf{W}_i$ applied to layer i, the second-order change in the loss can be approximated by the Fisher information matrix:

$$\Delta \mathcal{L} \approx \frac{1}{2} \operatorname{vec}(\Delta \mathbf{W}_i)^\top \mathbf{F}_i \operatorname{vec}(\Delta \mathbf{W}_i), \tag{3}$$

where $\operatorname{vec}(\cdot)$ denotes the vectorization operator. This follows from the second-order Taylor expansion:

$$\Delta \mathcal{L} \approx \frac{1}{2} \operatorname{vec}(\Delta \mathbf{W}_i)^\top \mathbf{H}_i \operatorname{vec}(\Delta \mathbf{W}_i), \tag{4}$$

by using $\mathbf{F}_i$ as a curvature substitute for $\mathbf{H}_i$. For negative log-likelihood losses, the population Fisher equals the expected Hessian:

$$\begin{aligned} \mathbf{F} &= \mathbb{E}\left[\nabla \log p_\theta \nabla \log p_\theta^\top\right] \\ &= \mathbb{E}\left[-\nabla^2 \log p_\theta\right] \end{aligned} \tag{5}$$

and coincides with the generalized Gauss–Newton matrix, making it a more stable PSD approximation to local curvature (Kunstner et al., 2019; Martens, 2020). Substituting Eq. (2) into Eq. (3) and using the identity $\operatorname{vec}(\mathbf{X})^\top (\mathbf{A} \otimes \mathbf{B}) \operatorname{vec}(\mathbf{X}) = \operatorname{Tr}(\mathbf{B}\mathbf{X}\mathbf{A}\mathbf{X}^\top)$, we obtain

$$\Delta \mathcal{L} \approx \frac{1}{2} \operatorname{Tr}\left(\boldsymbol{\Sigma}_\delta^i \Delta \mathbf{W}_i \boldsymbol{\Sigma}_h^i \Delta \mathbf{W}_i^\top\right). \tag{6}$$

To obtain a layer-level signal, FARSS considers a scaled perturbation along the weight direction, $\Delta \mathbf{W}_i = s \mathbf{W}_i$ with $s \ll 1$. Substituting into Eq. (6) yields:

$$\Delta \mathcal{L} \approx \frac{1}{2} s^2 \operatorname{Tr}\left(\boldsymbol{\Sigma}_\delta^i \mathbf{W}_i \boldsymbol{\Sigma}_h^i \mathbf{W}_i^\top\right). \tag{7}$$

FARSS therefore defines the Fisher importance of the i-th shallow linear layer as:

$$I_i = \frac{1}{m_i n_i} \operatorname{Tr}\left(\boldsymbol{\Sigma}_\delta^i \mathbf{W}_i \boldsymbol{\Sigma}_h^i \mathbf{W}_i^\top\right), \tag{8}$$

which measures the expected loss increase induced by a small relative perturbation of $\mathbf{W}_i$.

Inverse rank allocation. Given the shallow layers-set $\mathcal{E}$ (Section 3.1), FARSS allocates LoRA ranks within $\mathcal{E}$ by Fisher importance while keeping the total shallow layersrank Budget fixed. To reduce updates in sensitive layers, FARSS adopts an inverse allocation principle and defines:

$$\omega_i = \frac{1}{I_i + \epsilon}, \tag{9}$$

where ϵ is a small constant for numerical stability. Let r_{avg} denote the average rank in $\mathcal{E}$, giving the total budget:

$$R_{\text{total}} = r_{avg} \cdot |\mathcal{E}|. \tag{10}$$

The raw rank for the i-th shallow layersis

$$r_i^{\text{raw}} = \frac{R_{\text{total}}}{\sum_{j \in \mathcal{E}} \omega_j} \omega_i, \tag{11}$$

and the final rank is obtained by rounding and clipping:

$$r_i = \text{clip}(\text{round}(r_i^{\text{raw}}), r_{min}, r_{max}), \tag{12}$$

where r_{min} and r_{max} set the allowable rank range.

3.3 Task-Aware Singular Direction Initialization

After deciding where to adapt and how much capacity to allocate, FARSS reorders singular directions by task-derived statistics and initializes each adapter accordingly, yielding conservative shallow updates and effective deep updates. FARSS leverages task-derived covariance statistics and second-order information to guide initialization, aligning low-rank updates with directions that are most relevant for optimization.

Task-aware singular direction ranking. Consider a pre-trained linear weight $\mathbf{W} \in \mathbb{R}^{m \times n}$, FARSS begins by applying SVD:

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top, \tag{13}$$

where $\sigma_1 \geq \cdots \geq \sigma_k > 0$, $\mathbf{u}_i \in \mathbb{R}^m$, $\mathbf{v}_i \in \mathbb{R}^n$, and $k = \min(m, n)$.

To assess the influence of each singular direction, FARSS considers a rank-one perturbation $\Delta\mathbf{w}_i = \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$. Under the Fisher second-order approximation, the loss variation satisfies (as detailed in Appendix A.1.2):

$$q_i \propto \sigma_i^2 \lambda_i \gamma_i, \tag{14}$$

where $\lambda_i = \mathbf{v}_i^\top \mathbf{\Sigma}_h \mathbf{v}_i$ and $\gamma_i = \mathbf{u}_i^\top \mathbf{\Sigma}_\delta \mathbf{u}_i$ respectively characterize the importance of the i-th singular direction in the input activation space and the output gradient space. Based on this analysis, the importance score of the i-th singular direction is defined as:

$$s_i = \sigma_i^2 \lambda_i \gamma_i. \tag{15}$$

The score s_i estimates the expected loss sensitivity to a unit relative perturbation along the i-th singular direction under task-derived statistics. FARSS therefore uses higher-score directions to initialize adapters in deep layers $\mathcal{D}$, while using lower-score directions in shallow layers $\mathcal{E}$ to reduce interference with pre-trained representations.

FARSS reorder singular directions according to s_i in descending order, denoted by a permutation π such that $s_{\pi(1)} \geq s_{\pi(2)} \geq \cdots \geq s_{\pi(k)}$. Based on this ordering, FARSS constructs reordered SVD bases $\mathbf{U}'$, $\mathbf{V}'$, and $\mathbf{\Sigma}'$, which preserves the span of the original decomposition and yields the equivalent factorization $\mathbf{W} = \mathbf{U}'\mathbf{\Sigma}'\mathbf{V}'^\top$.

Initialization for Task Adaptation and Knowledge Preservation. Let $\mathcal{D}$ denote deep linear layers. For $l \in \mathcal{D}$, FARSS fix the rank to R and initialize:

$$\begin{aligned} \mathbf{B} &= \mathbf{U}'_{[:,:R]} \sqrt{\mathbf{\Sigma}'_{[:R]}}, \\ \mathbf{A} &= \sqrt{\mathbf{\Sigma}'_{[:R]}}\, \mathbf{V}'^\top_{[:R,:]}. \end{aligned} \tag{16}$$

For shallow linear layer $l \in \mathcal{E}$, FARSS instead uses the adaptive rank r_l determined in section 3.2 and initializes adapters using the least important singular directions under the same ordering:

$$\begin{aligned} \mathbf{B} &= \mathbf{U}'_{[:,-r_l:]} \sqrt{\mathbf{\Sigma}'_{[-r_l:]}}, \\ \mathbf{A} &= \sqrt{\mathbf{\Sigma}'_{[-r_l:]}}\, \mathbf{V}'^\top_{[-r_l:,:]}. \end{aligned} \tag{17}$$

This depth-aware initialization strategy ensures that shallow layers preserve knowledge, while deep layers focus on adapting to fine-tune task features.

4 Experiments

4.1 Experimental Setup

Models and tasks. We evaluate FARSS on natural language generation (NLG), natural language understanding (NLU), and commonsense reasoning tasks. For NLG, we fine-tune LLaMA2-7B (Touvron et al., 2023), Mistral-7B-v0.3, LLaMA-3-8B (Dubey et al., 2024), Gemma-2-9B (Team et al., 2024), and LLaMA2-13B (Touvron et al., 2023). For NLU, we fine-tune RoBERTa-base on the GLUE benchmark. For commonsense reasoning, we report results on both LLaMA2-7B and LLaMA3-8B to assess cross-model consistency.

Baselines. We compare FARSS with full fine-

(a) Math

Method	#Params	Trivia QA	NQ open	WebQS	GSM8k	Math	Avg	K	T	KT
LLaMA2-7B	-	52.51	14.99	5.86	-	-	-	-	-	-
Full fine-tuning	6738M	47.66	3.77	6.69	50.19	7.82	23.23	-	-	-
LoRA	320M	48.11	1.16	7.14	42.61	5.72	20.95	0.74	0.79	0.87
PiSSA	320M	41.50	2.52	6.55	**52.78**	7.82	22.23	0.69	1.03	0.80
MiLoRA	320M	48.48	4.18	6.05	39.58	6.16	20.89	0.74	0.79	0.76
LoRA-Null	320M	49.15	6.45	6.35	44.96	7.06	22.79	0.82	0.90	0.86
CorDA(KPM)	320M	44.03	**9.75**	5.61	49.05	7.68	23.22	0.82	0.98	0.93
KaSA	320M	**51.10**	8.75	6.99	32.45	4.40	20.74	**0.92**	0.60	0.73
FARSS (R=128)	271M	49.07	5.01	8.37	50.94	8.20	**24.32**	0.90	1.03	0.96
FARSS (R=153)	320M	48.23	4.27	**8.51**	51.63	**8.62**	24.25	0.89	**1.07**	**0.98**

(b) Code

Method	#Params	Trivia QA	NQ open	WebQS	HumanEval	MBPP	Avg	K	T	KT
LLaMA2-7B	-	52.51	14.99	5.86	-	-	-	-	-	-
Full fine-tuning	6738M	35.10	11.08	7.38	25.00	31.13	21.14	-	-	-
LoRA	320M	50.11	9.03	7.53	18.90	28.57	22.83	0.95	0.92	0.93
PiSSA	320M	47.94	10.28	8.86	18.29	30.42	23.16	1.04	0.93	0.98
MiLoRA	320M	49.35	11.50	6.84	16.46	27.78	22.39	0.96	0.84	0.90
LoRA-Null	320M	50.13	13.07	6.35	14.02	29.89	22.66	0.97	0.82	0.89
CorDA(KPM)	320M	48.60	9.42	9.25	18.90	**32.54**	23.74	1.04	0.95	1.00
KaSA	320M	50.12	**13.55**	6.10	14.63	25.4	21.96	0.97	0.76	0.85
FARSS (R=128)	285M	50.20	11.08	10.19	18.90	32.28	24.53	1.08	0.98	1.03
FARSS (R=144)	320M	**50.37**	10.97	**10.21**	**20.12**	31.83	**24.70**	**1.14**	**1.00**	**1.07**

(c) Instruction Following

Method	#Params	Trivia QA	NQ open	WebQS	IFEval				Avg	K	T	KT
					P-S	P-L	I-S	I-L				
LLaMA2-7B	-	52.51	14.99	5.86	-	-	-	-	-	-	-	-
Full fine-tuning	6738M	50.30	12.47	7.33	17.93	19.59	30.70	29.14	-	-	-	-
LoRA	320M	48.85	9.67	7.53	19.41	20.70	30.10	31.53	23.97	0.95	1.05	1.00
PiSSA	320M	47.86	9.28	8.37	19.59	21.44	31.29	33.45	24.47	0.99	1.09	1.03
MiLoRA	320M	49.22	11.39	7.19	19.04	21.44	30.10	32.49	24.41	0.97	1.06	1.02
LoRA-Null	320M	49.92	**14.38**	6.54	19.22	21.44	31.29	33.57	25.20	1.01	1.08	1.04
CorDA(KPM)	320M	**50.51**	13.52	7.43	19.41	21.07	32.13	33.81	25.41	1.04	1.09	1.07
KaSA	320M	50.36	12.08	6.50	18.11	19.59	29.62	30.94	23.89	0.96	1.01	0.98
FARSS (R=128)	285M	49.55	10.53	9.55	21.07	**23.66**	**32.85**	**35.37**	**26.08**	1.09	**1.17**	**1.13**
FARSS (R=144)	320M	49.47	10.61	**9.84**	**21.26**	**23.66**	32.49	34.77	26.01	**1.11**	1.16	**1.13**

Table 2: Knowledge-preserved adaptation results on LLaMA2-7B for (a) Math, (b) Code, and (c) instruction-following benchmarks. We report world knowledge scores (TriviaQA, NQ-open, WebQS), task metrics, their average (Avg), and the summary scores K, T, and KT across compared methods. (Complete results for other models, as well as all task settings, are provided in Appendix A.6.)

tuning and representative PEFT methods, including LoRA (Hu et al., 2022), PiSSA (Meng et al., 2024), MiLoRA (Wang et al., 2025), CorDA (Yang et al., 2024), KaSA (Wang et al., 2024a), and LoRA-Null (Tang et al., 2025).

Evaluation metrics. To assess the trade-off between task performance and world knowledge preservation, we report normalized metrics K, T, and their harmonic mean KT, which measure knowledge preservation, task performance relative to full fine-tuning, and their balance, respectively, and report averaged results across tasks where applicable.

All implementation details, including datasets, training settings, and metric definitions, are provided in Appendix A.2.

4.2 Main Results

Experimental Results on NLU Tasks and World Knowledge. Tables 2 report results on LLaMA-2-7B for Math, Code, and Instruction Following. For FARSS, the first seven layers are treated as shallow layers in Math, while the first five layers are treated as shallow layers in Code and Instruction

Following. Adaptive rank allocation uses $r_{min} = 1$, $r_{max} = 64$, and $r_{avg} = 50$ in all tasks. Full fine-tuning improves task benchmarks but degrades world knowledge on TriviaQA, NQ Open, and WebQS. PiSSA is competitive on task benchmarks but retains less world knowledge, whereas KaSA and CorDA better preserve knowledge scores but underperform on task benchmarks. We further report parameter-matched variants that only replace the deep-layer rank 128 with R (Math: R=153; Code and Instruction Following: R=144), keeping all other settings unchanged. These variants remain among the best and achieve the highest KT, indicating that a smaller parameter budget does not drive FARSS's gains. Overall, FARSS attains the best average and consistently the highest KT across settings, indicating a better balance between knowledge preservation and task adaptation.

On additional LLM backbones and task settings, including **Mistral-7B-v0.3**, **LLaMA-3-8B**, **LLaMA2-13B**, and **Gemma-2-9B**, FARSS exhibits the same trend, remaining best or tied best on **Avg** and **KT**, and thus offering the most effective balance between world knowledge preservation and task benchmark performance. Complete supporting results are provided in Appendix A.6.

Experimental Results on NLG Tasks. FARSS fine-tunes RoBERTa_{base} by applying adaptive rank allocation to the first six decoder layers for all linear modules except the classification head, using $r_{min} = 1$, $r_{max} = 64$, and $r_{avg} = 50$, while fixing remaining layers to rank 128. As shown in Table 3, FARSS attains the highest overall average and ranks among the top methods on most GLUE benchmarks. It improves the GLUE average over MiLoRA by 0.57 points, a relative gain of 0.67%, and reduces trainable parameters from 21M to 10M, a 52% reduction.

Experimental Results on commonsense reasoning. For commonsense reasoning, we adopt the hyperparameters in Table 6. Table 4 shows that FARSS achieves the best average results on both LLaMA2-7B and LLaMA3-8B while using fewer trainable parameters than all baselines, indicating stronger reasoning gains under a tighter parameter budget. FARSS delivers particularly strong performance on PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2021), and OpenBookQA (Mihaylov et al., 2018). We also report FARSS-init, which keeps only the initialization component and initializes adapters with the 32 lowest-scoring singular vectors. Despite its simplicity, FARSS-init attains the highest average on both models, demonstrating that the proposed initialization strategy provides a substantial portion of the overall improvement.

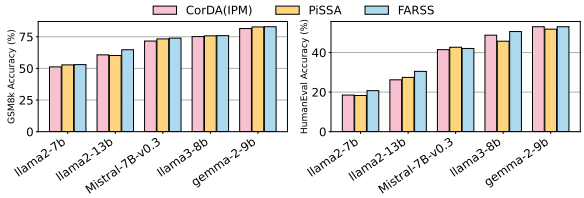


Figure 3: Results of FARSS initialization with rank 128 compared to PiSSA and CorDA (IPM) on GSM8K and HumanEval.

4.3 Ablation Study

We decompose FARSS into three factors: initialization, shallow layers split, and adaptive rank allocation, and isolate their effects under a unified pipeline, focusing on target task capability and world knowledge preservation. All ablations are conducted on LLaMA2-7B, except the initialization study, which was evaluated across multiple LLMs.

Initialization strategy. We retain only the initialization strategy and turn off shallow layers partitioning and adaptive rank allocation. Figure 3 shows consistent gains across LLMs on tasks such as GSM8K and HumanEval, with larger improvements at larger model scales. This suggests that the proposed initialization offers a better starting point and yields transferable benefits without additional structural components.

Adaptive Ranks Improve the Trade off. Figure 4 compares fixed shallow layers ranks with adaptive allocation under the same split. Fixed ranks exhibit strong sensitivity, resulting in oscillations in K, T, KT, and Average. In particular, rank 128 yields the weakest balance, with reduced K and degraded Avg, suggesting that excessive capacity in shallow layers causes over-updating. In contrast, adaptive allocation achieves the best balance, attaining the highest KT and stronger Avg while keeping T competitive, demonstrating the robustness of non-uniform allocation over fixed ranks.

shallow layers Split Boundary. We study the split boundary by varying the number of shallow layers that use adaptive rank allocation, while keeping deep layers at a fixed rank. Adaptive ranks are constrained by $r_{min} = 1$, $r_{max} = 64$, and an average budget $r_{avg} = 50$. As shown in Figure 5 and Figure 6, performance remains stable within

Method	#Params	SST-2	MNLI	MRPC	CoLA	QNLI	QQP	RTE	STS-B	Avg
Full fine-tuning	125M	95.18	88.46	88.53	61.82	92.88	91.57	74.73	90.30	85.43
LoRA	21M	92.55	82.10	88.24	63.22	89.49	83.58	52.71	91.05	80.37
PiSSA	21M	93.58	86.83	88.97	61.82	92.84	90.15	52.71	90.59	82.19
MiLoRA	21M	95.07	87.77	88.48	**63.31**	92.66	90.31	76.9	90.32	85.60
CorDA(IPM)	21M	93.46	86.55	84.80	53.83	92.35	**91.20**	62.09	89.09	81.67
KaSA	21M	94.61	87.36	89.71	60.09	92.79	91.07	75.46	90.49	85.20
FARSS (ours)	10M	**95.41**	**87.82**	**90.47**	63.22	**93.17**	90.08	**78.34**	**90.81**	**86.17**

Table 3: Performance Comparison on NLU Benchmarks. We compare FARSS with various PEFT baselines on the RoBERTa$_{base}$ model. The rank of LoRA, PiSSA, MiLoRA, CorDA and KaSA is 128.

Model	**Method(#Params)**	**BoolQ**	**PIQA**	**SIQA**	**HellaSwag**	**WinoGrande**	**ARC-e**	**ARC-c**	**OBQA**	**Avg.**
ChatGPT†	-	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LLaMA2-7B	LoRA (56M)	65.8	85.9	79.6	92.7	74.4	85.7	66.2	70.1	77.6
	PiSSA (56M)	71.0	82.4	79.6	90.4	67.5	77.2	56.7	65.6	73.8
	MiLoRA (56M)	73.8	86.3	80.4	89.1	73.2	**86.4**	67.2	71.4	78.5
	CorDA(IPM) (56M)	68.3	83.6	79.3	93.3	64.6	84.4	65.2	67.2	75.7
	KaSA (56M)	70.6	83.9	80.3	93.7	64.1	**86.4**	64.3	67.4	76.4
	FARSS (50M)	74.3	85.8	80.6	93.2	73.2	85.6	66.4	71.0	78.8
	FARSS-init (56M)	**74.5**	**86.4**	**80.7**	**94.1**	**75.0**	86.2	**67.3**	**71.6**	**79.5**
LLaMA3-8B	LoRA (57M)	76.2	89.9	81.8	**96.8**	78.1	91.0	72.1	78.2	83.0
	PiSSA (57M)	66.8	79.5	75.9	73.4	58.5	67.7	49.0	55.7	65.8
	MiLoRA (57M)	74.3	85.7	79.6	94.3	74.7	88.4	71.1	72.8	80.1
	CorDA(IPM) (57M)	75.0	89.0	82.4	96.3	76.6	91.5	**78.8**	78.2	83.5
	KaSA (57M)	72.9	87.7	81.3	95.8	72.6	90.7	76.8	77.8	81.9
	FARSS (47M)	76.2	**90.5**	82.0	96.2	78.9	90.7	75.7	80.4	83.8
	FARSS-init (57M)	**76.6**	**90.5**	**82.6**	**96.8**	**79.5**	**91.6**	77.6	**81.6**	**84.6**

Table 4: Performance comparison of LLaMA2-7B and LLaMA3-8B with different adaptation methods on eight commonsense reasoning datasets. The symbol † indicates that the results are taken from (Wang et al., 2025).

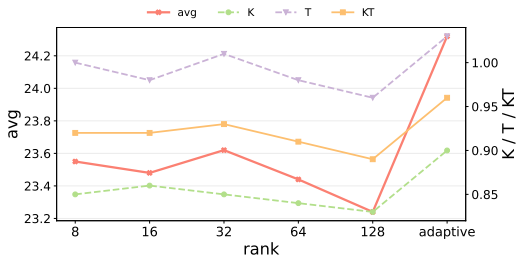


Figure 4: Sensitivity to shallow layersrank choices and comparison with adaptive rank allocation in the first seven layers, measured by Avg, K, T, and KT.

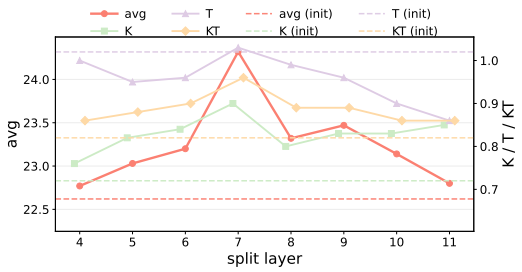


Figure 5: Split boundary sensitivity on Math. The first N decoder layers use adaptive ranks, while deep layers are fixed.

a favorable range. Using too few shallow layers weakens knowledge preservation and reduces the overall score, whereas using too many shallow layers limits task gains and lowers the overall score.

Adaptive rank allocation settings. Table 5 shows that a moderate average rank budget together with a restrained maximum rank yields the most favorable balance, improving task benchmarks while preserving world knowledge more consistently than either overly tight or overly loose allocations.

5 Conclusion

In this paper, we propose FARSS, a LoRA fine-tuning framework that assigns adaptive ranks to shallow layers, fixes ranks in deep layers, and initializes adapters along optimization-favorable singular directions to better preserve world knowledge. Experimental results across a range of tasks, including Math, Code, and Instruction Following, demonstrate that FARSS consistently outperforms existing PEFT methods in terms of both knowledge preservation and task performance. Furthermore, the method remains robust across different model backbones and parameter budgets, achieving the optimal overall balance between preserving world knowledge and adapting to tasks. These findings highlight the promise of FARSS for improving the stability and efficiency of large language model fine-tuning.

Limitations

In this paper, we evaluate FARSS on Math, code, and instruction-following tasks. An interesting direction is to extend the framework to broader adaptation settings, such as multimodal and generative models, where both efficient fine-tuning and knowledge preservation are essential. We also plan to study more diverse training objectives and evaluation protocols to further characterize the approach's generality.

Ethics Statement

The proposed method focuses on improving the parameter efficiency and stability of fine-tuning large language models. As with other fine-tuning techniques, it can be applied to adapt models toward a wide range of target task behaviors, including potentially harmful or misleading content if misused. These risks are not introduced by our method itself but are inherent to the deployment of generative models. We encourage responsible use of fine-tuning techniques in accordance with existing safety guidelines, and emphasize that appropriate data curation and deployment safeguards remain essential.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Mark Chen. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Q Chen and et al. 2024. Robust adapter tuning against forgetting. *AAAI*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. 2023a. Sparse low-rank adaptation of pre-trained language models. *arXiv preprint arXiv:2311.11696*.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, and 1 others. 2023b. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, and 1 others. 2024. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Chenghao Fan, Zhenyi Lu, Sichen Liu, Chengfeng Gu, Xiaoye Qu, Wei Wei, and Yu Cheng. 2025. Make lora great again: Boosting lora with adaptive singular values and mixture-of-experts optimization alignment. *arXiv preprint arXiv:2502.16894*.

Jiancheng Gu, Jiabin Yuan, Jiyuan Cai, Xianfa Zhou, and Lili Fan. 2025. La-lora: Parameter-efficient fine-tuning with layer-wise adaptive low-rank adaptation. *Neural Networks*, page 108095.

Haiyang Guo, Fanhu Zeng, Ziwei Xiang, Fei Zhu, Da-Han Wang, Xu-Yao Zhang, and Cheng-Lin Liu. 2025. Hide-llava: Hierarchical decoupling for continual instruction tuning of multimodal large language model. *arXiv preprint arXiv:2503.12941*.

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

Haonan He, Peng Ye, Yuchen Ren, Yuan Yuan, Luyang Zhou, Shucun Ju, and Lei Chen. 2025a. Gora: Gradient-driven adaptive low rank adaptation. *arXiv preprint arXiv:2502.12171*.

Haoze He, Juncheng B Li, Xuan Jiang, and Heather Miller. 2025b. Smt: Fine-tuning large language models with sparse matrices. In *The Thirteenth International Conference on Learning Representations*.

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

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

Jerry Yao-Chieh Hu, Maojiang Su, Zhao Song, Han Liu, and 1 others. 2025. Computational limits of low-rank adaptation (lora) fine-tuning for transformer models. In *The Thirteenth International Conference on Learning Representations*.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 5254–5276.

Weizhong Huang, Yuxin Zhang, Xiawu Zheng, Yang Liu, Jing Lin, Yiwu Yao, and Rongrong Ji. 2025. Dynamic low-rank sparse adaptation for large language models. *arXiv preprint arXiv:2502.14816*.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, and 1 others. 2024. Exploring concept depth: How large language models acquire knowledge at different layers? *CoRR*.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, and 1 others. 2025. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In *Proceedings of the 31st international conference on computational linguistics*, pages 558–573.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Damjan Kalajdzievski. 2024. Scaling laws for forgetting when fine-tuning large language models. *arXiv preprint arXiv:2401.05605*.

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

Frederik Kunstner, Philipp Hennig, and Lukas Balles. 2019. Limitations of the empirical fisher approximation for natural gradient descent. *Advances in neural information processing systems*, 32.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024a. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

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

Zequan Liu, Jiawen Lyn, Wei Zhu, Xing Tian, and Yvette Graham. 2024b. Alora: Allocating low-rank adaptation for fine-tuning large language models. *arXiv preprint arXiv:2403.16187*.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. *arXiv preprint arXiv:2106.04489*.

James Martens. 2020. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76.

James Martens and Roger Grosse. 2015. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417. PMLR.

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

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

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2025. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72.

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

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training.

Yehonathan Refael, Jonathan Svirsky, Boris Shustin, Wasim Huleihel, and Ofir Lindenbaum. 2024. Adarankgrad: Adaptive gradient-rank and moments for memory-efficient llms training and fine-tuning. *arXiv preprint arXiv:2410.17881*.

Weijieying Ren, Xinlong Li, Lei Wang, Tianxiang Zhao, and Wei Qin. 2024. Analyzing and reducing catastrophic forgetting in parameter efficient tuning. *arXiv preprint arXiv:2402.18865*.

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

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*.

Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. 2025. Transformer layers as painters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25219–25227.

Pengwei Tang, Yong Liu, Dongjie Zhang, Xing Wu, and Debing Zhang. 2025. Lora-null: Low-rank adaptation via null space for large language models. *arXiv preprint arXiv:2503.02659*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2023. Dylora: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3274–3287.

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

Fan Wang, Juyong Jiang, Chansung Park, Sunghun Kim, and Jing Tang. 2024a. Kasa: Knowledge-aware singular-value adaptation of large language models. *arXiv preprint arXiv:2412.06071*.

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

Haoyu Wang, Tianci Liu, Ruirui Li, Monica Xiao Cheng, Tuo Zhao, and Jing Gao. 2024b. Roselora: Row and column-wise sparse low-rank adaptation of pre-trained language model for knowledge editing and fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 996–1008.

Yifeng Xiong and Xiaohui Xie. 2025. Oplora: Orthogonal projection lora prevents catastrophic forgetting during parameter-efficient fine-tuning. *arXiv preprint arXiv:2510.13003*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*.

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. 2024. Corda: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. *Advances in Neural Information Processing Systems*, 37:71768–71791.

Kai Yao, Penglei Gao, Lichun Li, Yuan Zhao, Xiaofeng Wang, Wei Wang, and Jianke Zhu. 2024. Layer-wise importance matters: Less memory for better performance in parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2410.11772*.

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

Fanhu Zeng, Haiyang Guo, Fei Zhu, Li Shen, and Hao Tang. 2025. Robustmerge: Parameter-efficient model merging for mllms with direction robustness. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. 2023a. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303*.

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

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. 2024. Galore: Memory-efficient llm training by gradient low-rank projection. In *International Conference on Machine Learning*, pages 61121–61143. PMLR.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. 2024. Opencodeinterpreter: Integrating code generation with execution and refinement. *arXiv preprint arXiv:2402.14658*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Didi Zhu, Zhongyisun Sun, Zexi Li, Tao Shen, Ke Yan, Shouhong Ding, Chao Wu, and Kun Kuang. 2024. Model tailor: Mitigating catastrophic forgetting in multi-modal large language models. In *International Conference on Machine Learning*, pages 62581–62598. PMLR.

A Appendix

A.1 Proof

A.1.1 K-FAC Approximation of the Fisher Information Matrix

Consider a linear layer

$$\mathbf{y} = \mathbf{W}\mathbf{h}, \tag{18}$$

where $\mathbf{W} \in \mathbb{R}^{m\times n}$ and $\mathbf{h} \in \mathbb{R}^n$. Let the loss gradient with respect to the layer output be

$$\boldsymbol{\delta} = \frac{\partial \mathcal{L}}{\partial \mathbf{y}} \in \mathbb{R}^m. \tag{19}$$

The gradient of the loss with respect to the weight matrix is given by

$$\mathbf{G} = \frac{\partial \mathcal{L}}{\partial \mathbf{W}} = \boldsymbol{\delta}\mathbf{h}^\top \in \mathbb{R}^{m\times n}. \tag{20}$$

Let $\mathrm{vec}(\cdot)$ denote the vectorization operator. Using the identity $\mathrm{vec}(\mathbf{u}\mathbf{v}^\top) = \mathbf{v} \otimes \mathbf{u}$, where $\otimes$ denotes the Kronecker product, we obtain

$$\mathbf{g} \triangleq \mathrm{vec}(\mathbf{G}) = \mathrm{vec}(\boldsymbol{\delta}\mathbf{h}^\top) = \mathbf{h} \otimes \boldsymbol{\delta}. \tag{21}$$

The Fisher information matrix is defined as

$$\mathbf{F} \triangleq \mathbb{E}\Big[\mathbf{g}\mathbf{g}^\top\Big] = \mathbb{E}\Big[(\mathbf{h} \otimes \boldsymbol{\delta})(\mathbf{h} \otimes \boldsymbol{\delta})^\top\Big]. \tag{22}$$

Applying the multiplication rule of the Kronecker product yields

$$\mathbf{F} = \mathbb{E}\Big[(\mathbf{h}\mathbf{h}^\top) \otimes (\boldsymbol{\delta}\boldsymbol{\delta}^\top)\Big]. \tag{23}$$

Under the K-FAC independence assumption (Martens and Grosse, 2015), we approximate the joint expectation as the Kronecker product of marginal second-order statistics:

$$\boldsymbol{\Sigma}_h \triangleq \mathbb{E}[\mathbf{h}\mathbf{h}^\top], \qquad \boldsymbol{\Sigma}_\delta \triangleq \mathbb{E}[\boldsymbol{\delta}\boldsymbol{\delta}^\top]. \tag{24}$$

The Fisher information matrix is thus approximated as

$$\mathbf{F} \approx \boldsymbol{\Sigma}_h \otimes \boldsymbol{\Sigma}_\delta. \tag{25}$$

A.1.2 From Kronecker-Factored Fisher to the Quadratic Form of Matrix Perturbations

Consider a small perturbation $\Delta\mathbf{W}$ applied to the weight matrix. The second-order change in the loss can be approximated as

$$\Delta^2\mathcal{L} \approx \frac{1}{2}\,\mathrm{vec}(\Delta\mathbf{W})^\top \mathbf{F}\,\mathrm{vec}(\Delta\mathbf{W}). \tag{26}$$

Substituting the K-FAC approximation in Eq. (25) gives

$$Q(\Delta\mathbf{W}) \triangleq \mathrm{vec}(\Delta\mathbf{W})^\top(\boldsymbol{\Sigma}_h \otimes \boldsymbol{\Sigma}_\delta)\mathrm{vec}(\Delta\mathbf{W}). \tag{27}$$

Using the identity

$$\mathrm{vec}(\mathbf{X})^\top(\mathbf{C} \otimes \mathbf{A})\mathrm{vec}(\mathbf{X}) = \mathrm{Tr}(\mathbf{A}\mathbf{X}\mathbf{C}\mathbf{X}^\top), \tag{28}$$

we obtain

$$Q(\Delta\mathbf{W}) \approx \mathrm{Tr}\Big(\boldsymbol{\Sigma}_\delta \Delta\mathbf{W}\boldsymbol{\Sigma}_h\Delta\mathbf{W}^\top\Big). \tag{29}$$

Next, we perform singular value decomposition of the weight matrix

$$\mathbf{W} = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i\mathbf{v}_i^\top, \quad \mathbf{u}_i \in \mathbb{R}^m,\ \mathbf{v}_i \in \mathbb{R}^n. \tag{30}$$

We define

$$\lambda_i = \mathbf{v}_i^\top \boldsymbol{\Sigma}_h \mathbf{v}_i, \qquad \gamma_i = \mathbf{u}_i^\top \boldsymbol{\Sigma}_\delta \mathbf{u}_i. \tag{31}$$

Ignoring the coupling between different singular directions, we consider the rank-one perturbation

$$\Delta\mathbf{w}_i = \sigma_i \mathbf{u}_i\mathbf{v}_i^\top. \tag{32}$$

Substituting Eq. (32) into Eq. (29) yields

$$Q(\Delta\mathbf{w}_i) \approx \sigma_i^2 \mathrm{Tr}\Big(\boldsymbol{\Sigma}_\delta \mathbf{u}_i\mathbf{v}_i^\top \boldsymbol{\Sigma}_h \mathbf{v}_i\mathbf{u}_i^\top\Big). \tag{33}$$

Using the cyclic invariance of the trace, this expression simplifies to

$$\begin{aligned} Q(\Delta\mathbf{w}_i) &\approx \sigma_i^2(\mathbf{u}_i^\top \boldsymbol{\Sigma}_\delta \mathbf{u}_i)(\mathbf{v}_i^\top \boldsymbol{\Sigma}_h \mathbf{v}_i) \\ &= \sigma_i^2 \cdot \gamma_i \cdot \lambda_i. \end{aligned} \tag{34}$$

Therefore, the Fisher-based importance score of the i-th singular direction is defined as

$$s_i = \sigma_i^2 \gamma_i \lambda_i, \tag{35}$$

which measures the relative sensitivity of the loss to perturbations along that direction.

A.2 More Implementation Details

All experiments run on NVIDIA H800 80GB GPUs with a shared training pipeline. For PEFT baselines, adapters are inserted into the same target linear layers and key settings are aligned, including the rank budget and LoRA hyperparameters such as rank, alpha, and dropout.

A.2.1 Fientuning on Math, Code, and Instruction Following

Optimization is performed with AdamW using a learning rate of 2e-5 and a linear learning rate schedule with a 0.03 warmup ratio; we do not apply weight decay. We set lora_dropout to 0.05, use BFloat16 precision, a LoRA rank (r) and alpha (α) of 128, and an adequate batch size of 16. For task-specific initialization, we randomly sample 256 examples from the corresponding training dataset to construct the initialization signals. During fine-tuning, we randomly sample 100,000 training instances and train for one epoch, where the loss is computed only on the response tokens. World knowledge evaluation is conducted on TriviaQA, NQ Open, and WebQS using lm_eval version 0.4.9.1. All runs use PyTorch 2.8.0+cu128, Transformers 4.57.0, and PEFT 0.17.1.

A.3 Evaluation Metrics

To quantify the trade-off between world knowledge preservation and task performance, we report three normalized scores K, T, and KT. Let $\{W_j\}_{j=1}^{M}$ and $\{W_j^{(0)}\}_{j=1}^{M}$ denote the post-finetuning and pre-finetuning results on the world knowledge benchmarks, respectively. We define

$$K = \frac{1}{M}\sum_{j=1}^{M}\frac{W_j}{W_j^{(0)}}. \tag{36}$$

The score K reflects how well world knowledge is preserved after fine-tuning. Let $\{S_i\}_{i=1}^{N}$ denote the post-finetuning results on task benchmarks, and $\{S_i^{(\mathrm{FT})}\}_{i=1}^{N}$ denote the corresponding results of full fine-tuning. We define

$$T = \frac{1}{N}\sum_{i=1}^{N}\frac{S_i}{S_i^{(\mathrm{FT})}}. \tag{37}$$

whereas T reflects how close task performance is to that of full fine-tuning. Finally, we combine the two scores using their harmonic mean:

$$KT = \frac{2 \cdot K \cdot T}{K + T}. \tag{38}$$

A larger KT indicates a better overall balance, favoring methods that retain world knowledge while maintaining strong task performance.

A.4 Datasets

The world knowledge is evaluated by the exact match scores (%) on TriviaQA (Joshi et al., 2017), NQ Open (Lee et al., 2019), and WebQS (Berant et al., 2013). For NLG, Math ability is trained on MetaMathQA (Yu et al., 2023) and tested on GSM8K (Cobbe et al., 2021) and Math (Yu et al., 2023). Code is trained on CodeFeedback (Zheng et al., 2024) and evaluated on HumanEval (Chen, 2021) and MBPP (Austin et al., 2021). Instruction following is trained on WizardLM-Evol-Instruct (Xu et al., 2024) and evaluated on IFEval (Zhou et al., 2023), reporting P-S (Prompt-strict), P-L (Prompt-loose), I-S (Instruction-strict), and I-L (Instruction-loose). For NLU, RoBERTa$_{\text{base}}$ (Liu et al., 2019) is evaluated on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018). For commonsense reasoning, LLaMA2-7B and LLaMA3-8B are trained on Commonsense170K (Hu et al., 2023) and evaluated on the individual test sets of its constituent datasets.

A.4.1 GLUE Benchmark

To ensure fair comparison across Full fine-tuning, LoRA, Pissa, Milora, KaSA, CorDA and FARSS in the GLUE benchmark, we implement all methods under the same training and evaluation settings. we integrate all methods modules into the Query, Value, ttention Output, and first Fully Connected weight matrices of the transformer model.The AdamW optimizer is used with a batch size of 64 and a learning rate of 4e-4 for 10 epochs, following a linear learning rate schedule. The max token length is set as 512. The rank of methods is 128. All methods are trained on a single NVIDIA H800 80G GPU.

A.4.2 Commonsense Reasoning

For commonsense reasoning, we follow a unified training and evaluation protocol to ensure fair comparisons across LoRA, PiSSA, MiLoRA, and CorDA, and keep all configurations identical unless a baseline introduces method-specific hyperparameters. For KaSA, we adopt its dedicated regularization coefficients and tune them per backbone: on LLaMA2-7B, KaSA uses $\beta = 1 \times 10^{-2}$ and $\gamma = 1 \times 10^{-4}$; on LLaMA3-8B, KaSA uses $\beta = 1 \times 10^{-4}$ and $\gamma = 1 \times 10^{-3}$. All other settings, including data splits, optimization, and evaluation, remain matched across methods.

A.5 Ablation Study

This section provides additional ablation results that complement the main text. Figure 6 reports the sensitivity to the shallow layers split boundary on the Code task, where the first N decoder layers

Adaptive rank setting	NQ Open	GSM8K	Avg
(1, 64, 128)	2.69	50.72	26.71
(1, 32, 64)	2.52	49.73	26.13
(1, 16, 32)	2.60	49.73	26.17
(1, 50, 64)	**5.01**	**50.94**	**26.98**
(1, 16, 64)	2.35	49.81	26.08

Table 5: Sensitivity to adaptive rank allocation settings on LLaMA2-7B. The first seven decoder layers are treated as shallow layers for adaptive allocation, while deep layers are fixed. The tuple $(r_{min}, r_{avg}, r_{max})$ specifies the minimum rank, the average rank budget, and the maximum rank used for adaptive allocation in the shallow layers.

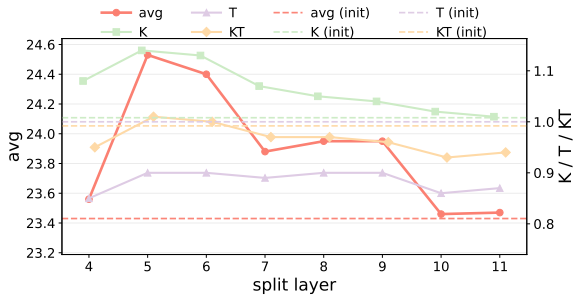


Figure 6: Split boundary sensitivity on Code. The first N decoder layers use adaptive ranks, while deep layers are fixed.

Hyperparameters(FARSS)	LLaMA2-7B	LLaMA3-8B
shallow layerssplit	5	7
r_{min}	1	
r_{max}	16	12
r_{avg}	24	16
Rank r	32	
α	64	
Dropout	0.05	
Optimizer	AdamW	
LR	3e-4	
LR Scheduler	Linear	
Batch size	16	
Warmup Steps	100	
Epochs	3	
Where	Q, K, V, Up, Down	

Table 6: Hyperparameter configurations of FARSS for LLaMA2-7B and LLaMA3-8B on commonsense reasoning tasks.

adopt adaptive rank allocation and deep layers remain fixed. Table 5 further investigates the choice of adaptive rank hyperparameters under the same partitioning scheme.

A.6 More Experimental Results

We present additional experimental results and analyses to further validate the effectiveness of our method across diverse backbones and domains. Unless otherwise specified, all adaptive rank allocation experiments adopt a unified configuration with $r_{min} = 1$, $r_{max} = 64$, and $r_{avg} = 50$. For layer partitioning, we use model-specific shallow layer cutoffs, following the intuition that shallower layers contribute more to general/world knowledge, while deep layers are more responsible for task specialization. Concretely, for Mistral-7B-v0.3, we treat the first five layers as shallow layers for Math and Code, and the first seven layers as shallow layers for Instruction Following. For LLaMA3-8B, the first nine layers are considered shallow layers for both Math and Code. For Gemma-2-9B, we use the first 11 layers as shallow layers for both Code and Instruction Following. For LLaMA2-13B, we set the first 11 layers as shallow layers for both Math and Code. These settings are applied consistently across methods to ensure a fair comparison, and detailed results are summarised in Tables 7, Tables 8, and Tables 9. To enable a like-for-like comparison, we additionally report results under matched trainable-parameter budgets within each backbone family.

Across all benchmarks, a consistent pattern emerges. Methods that prioritise task performance tend to incur larger degradation in world knowledge preservation, whereas approaches that focus on knowledge preservation often exhibit smaller gains on the target tasks. In particular, full fine-tuning frequently achieves the strongest task scores while reducing knowledge-oriented metrics, resulting in a less favorable overall balance. Conversely, CorDA and KaSA better preserve world knowledge in several settings. Still, their improvements on Code and Instruction Following are often limited, suggesting an inherent trade-off in their optimisation constraints. In contrast, our method achieves a more favorable compromise: it improves target-task capability while maintaining competitive knowledge scores, as reflected in consistently strong average performance and the balance metric KT across models.

More importantly, our advantage becomes most evident when considering K, T, and KT jointly. The tables show that several baselines can excel in either task performance or knowledge preservation in isolation, but struggle to maintain both simultaneously. For example, PiSSA tends to inflate task metrics on Code for Mistral while weakening knowledge preservation. In contrast, CorDA and KaSA more reliably preserve knowledge but deliver smaller gains in code and instruction following. Under matched parameter budgets, our method

Model	Method	#Params	NQ open	TriviaQA	WebQS	GSM8K	Math	Avg	K	T	KT
Mistral-7B-v0.3	origin	-	20.53	60.62	8.46	-	-	-	-	-	-
	Full fine-tuning	7248M	-	-	-	67.55	18.90	-	-	-	-
	CorDA	336M	**16.76**	45.40	6.35	72.02	21.76	32.46	0.77	1.11	0.91
	PiSSA	336M	10.08	48.62	9.40	73.31	**22.54**	32.79	0.80	**1.14**	0.94
	MiLoRA	336M	13.27	48.72	10.24	69.98	20.98	32.64	0.89	1.07	0.97
	LoRA	336M	10.33	47.41	12.50	70.81	21.48	32.51	0.92	1.09	1.00
	LoRA-Null	336M	12.61	50.50	10.98	68.76	20.94	32.76	0.92	1.06	0.98
	KaSA	336M	12.39	52.97	11.09	62.70	17.42	31.31	0.93	0.92	0.93
	FARSS (R=128)	300M	13.07	53.81	12.30	73.16	21.70	34.81	0.99	1.12	1.05
	FARSS (R=152)	336M	14.18	**54.53**	**14.96**	**74.98**	21.36	**36.00**	**1.12**	1.12	**1.12**
LLaMA-3-8B	origin	-	13.71	63.32	13.73	-	-	-	-	-	-
	Full fine-tuning	8366M	5.57	55.17	7.92	77.78	26.44	-	-	-	-
	CorDA	336M	8.64	60.00	10.78	75.66	25.20	36.06	0.79	0.96	0.87
	PiSSA	336M	5.18	47.07	5.51	75.73	24.58	31.61	0.51	0.95	0.66
	MiLoRA	336M	8.25	59.35	12.16	71.27	23.18	34.84	0.81	0.90	0.85
	LoRA	336M	8.35	60.71	12.19	70.13	23.30	34.94	0.82	0.89	0.85
	LoRA-Null	336M	8.61	61.32	12.55	72.07	23.10	35.53	0.84	0.90	0.87
	KaSA	336M	9.35	**61.89**	**13.44**	65.28	20.88	34.17	**0.88**	0.81	0.85
	FARSS (R=128)	276M	**10.75**	60.37	10.38	74.83	24.96	36.26	0.83	0.95	**0.89**
	FARSS (R=155)	336M	10.72	58.41	9.97	**76.34**	**26.26**	**36.34**	0.81	**0.99**	**0.89**
LLaMA2-13B	origin	-	23.66	60.86	11.42	-	-	-	-	-	-
	Full fine-tuning	13016M	0.25	0.03	0.00	64.67	15.62	16.11	-	-	-
	CorDA	501M	**21.16**	56.14	4.63	61.03	11.58	**30.91**	0.74	0.84	0.79
	PiSSA	501M	11.11	41.46	4.04	60.19	11.14	25.59	0.50	0.82	0.62
	MiLoRA	501M	19.53	57.57	7.78	50.18	8.26	28.66	0.82	0.65	0.73
	LoRA	501M	17.01	58.24	8.51	52.84	7.98	28.92	0.81	0.66	0.73
	LoRA-Null	501M	19.47	54.59	6.74	55.26	8.92	29.00	0.77	0.71	0.74
	KaSA	501M	20.80	**59.30**	**9.10**	44.12	6.80	28.02	**0.88**	0.56	0.68
	FARSS (R=128)	400M	13.24	53.27	8.37	62.32	11.22	29.68	0.72	0.84	0.78
	FARSS (R=164)	501M	14.90	52.85	7.87	**62.77**	**12.18**	30.11	0.73	**0.88**	**0.80**

Table 7: Knowledge-preserved adaptation results on Math benchmarks (GSM8K and Math) with world knowledge scores (NQ-open, TriviaQA, WebQS) and summary metrics K, T, and KT.

consistently delivers the most robust balance across backbones and domains, achieving the highest or comparable KT while maintaining strong averages. Overall, these results support our design goal: guiding adaptation toward task-relevant directions without excessively drifting from pre-trained representations that encode world knowledge.

Model	Method	#Params	NQ open	TriviaQA	WebQS	HumanEval	MBPP	Avg	K	T	KT
Mistral-7B-v0.3	origin	-	20.53	60.62	8.46	-	-	-	-	-	-
	Full fine-tuning	7248M	17.40	49.57	14.47	37.80	47.35	33.32	-	-	-
	CorDA	336M	17.42	48.19	7.38	41.46	50.26	32.94	0.84	1.08	0.94
	PiSSA	336M	13.91	42.73	11.22	**42.68**	**52.38**	32.58	0.90	**1.12**	1.00
	MiLoRA	336M	20.00	61.67	14.37	35.37	46.03	35.49	1.23	0.95	1.07
	LoRA	336M	17.81	60.71	15.16	33.54	47.88	35.02	1.22	0.95	1.07
	LoRA-Null	336M	20.00	60.90	13.14	34.76	48.94	35.55	1.18	0.98	1.07
	KaSA	336M	21.39	61.97	13.09	32.93	46.30	35.14	1.20	0.92	1.05
	FARSS (R=128)	300M	17.92	60.14	14.37	38.41	51.32	36.43	1.19	1.05	1.11
	FARSS (R=152)	336M	**21.40**	**61.99**	**19.69**	39.02	48.12	**38.04**	**1.46**	1.02	**1.21**
LLaMA-3-8B	origin	-	13.71	63.32	13.73	-	-	-	-	-	-
	Full fine-tuning	8366M	16.23	61.40	13.83	51.83	58.73	40.40	-	-	-
	CorDA	336M	15.35	62.91	14.17	46.95	57.24	39.32	1.05	0.94	**0.99**
	PiSSA	336M	15.82	60.10	9.99	45.73	53.44	37.02	0.94	0.90	0.92
	MiLoRA	336M	14.04	62.52	12.80	43.90	55.56	37.76	0.98	0.90	0.94
	LoRA	336M	16.26	60.16	13.09	46.95	55.56	38.40	1.03	0.93	0.98
	LoRA-Null	336M	14.18	62.30	12.84	44.51	56.88	38.14	0.98	0.91	0.95
	KaSA	336M	15.57	**63.38**	**15.31**	42.07	51.06	37.48	**1.08**	0.84	0.95
	FARSS (R=128)	276M	**16.37**	62.41	12.30	**49.39**	**57.41**	**39.58**	1.03	**0.97**	**0.99**
	FARSS (R=155)	336M	14.34	62.54	12.84	48.78	54.89	38.72	0.99	0.94	0.97
LLaMA2-13B	origin	-	23.66	60.86	11.42	-	-	-	-	-	-
	Full fine-tuning	13016M	23.82	60.22	9.94	29.88	39.95	32.76	-	-	-
	CorDA	501M	24.02	57.98	11.61	26.22	39.15	31.80	0.99	0.93	0.96
	PiSSA	501M	20.22	55.55	9.94	27.44	37.57	30.14	0.88	0.93	0.90
	MiLoRA	501M	24.79	59.98	9.99	26.22	35.71	31.34	0.97	0.89	0.93
	LoRA	501M	24.29	60.70	10.33	25.00	35.98	31.26	0.98	0.87	0.92
	LoRA-Null	501M	23.66	58.90	9.20	25.44	39.15	31.27	0.92	0.92	0.92
	KaSA	501M	**25.51**	**61.34**	**11.66**	17.68	32.80	29.80	**1.04**	0.71	0.84
	FARSS (R=128)	421M	23.91	58.70	11.42	26.22	**39.68**	31.99	0.99	0.94	0.96
	FARSS (R=155)	501M	24.74	57.35	10.78	**34.15**	37.84	**32.97**	0.98	**1.05**	**1.01**
Gemma-2-9B	origin	-	26.12	68.02	18.95	-	-	-	-	-	-
	Full fine-tuning	9242M	25.12	64.95	22.10	53.05	48.68	42.78	-	-	-
	CorDA	432M	**26.51**	68.15	22.15	53.05	58.47	45.67	1.04	1.10	1.07
	PiSSA	432M	25.65	67.98	24.26	51.83	**59.26**	45.80	1.07	1.10	1.08
	MiLoRA	432M	26.37	**68.45**	24.41	50.00	56.35	45.12	1.08	1.05	1.06
	LoRA	432M	26.32	68.34	23.72	51.22	56.35	45.19	1.07	1.06	1.06
	LoRA-Null	432M	26.34	68.15	22.29	49.39	56.88	44.61	1.04	1.05	1.05
	KaSA	432M	23.99	65.69	17.22	46.95	56.35	42.04	0.92	1.02	0.97
	FARSS (R=128)	363M	25.82	67.67	**25.39**	54.88	56.88	46.13	**1.09**	1.10	1.09
	FARSS (R=155)	432M	25.82	68.30	24.90	**55.49**	**59.26**	**46.75**	1.08	**1.13**	**1.11**

Table 8: Knowledge-preserved adaptation results on Code benchmarks (HumanEval and MBPP) with world knowledge scores (NQ-open, TriviaQA, WebQS) and summary metrics K, T, and KT.

Model	Method	#Params	TriviaQA	NQ open	WebQS	IFEval				Avg	K	T	KT
						P-S	P-L	I-S	I-L				
Mistral-7B-v0.3	origin	-	20.53	60.62	8.46	-	-	-	-	-	-	-	-
	Full fine-tuning	7248M	10.61	15.25	10.53	30.87	33.46	42.33	44.36	26.77	-	-	-
	CorDA	336M	12.69	44.64	7.82	19.41	21.63	32.85	34.65	24.81	0.76	0.71	0.74
	PiSSA	336M	16.62	54.59	16.04	27.17	29.39	39.93	42.69	32.35	1.20	0.92	1.04
	MiLoRA	336M	20.39	61.10	11.17	21.07	22.92	32.61	34.65	29.13	1.11	0.73	0.88
	LoRA	336M	18.31	59.80	11.07	22.37	25.32	35.85	38.73	30.21	1.06	0.81	0.92
	LoRA-Null	336M	19.81	58.89	12.75	19.04	21.44	31.53	33.93	28.20	1.15	0.70	0.87
	KaSA	336M	**22.35**	**61.91**	12.45	18.67	21.44	32.97	35.73	29.36	1.19	0.71	0.89
	FARSS (R=128)	300M	20.58	56.89	**18.45**	28.84	32.72	41.01	44.48	34.71	**1.37**	0.98	1.14
	FARSS (R=144)	336M	19.25	55.95	18.36	**29.39**	**34.20**	**41.49**	**46.04**	**34.95**	1.34	**1.00**	**1.15**
LLaMA2-13B	origin	-	23.66	60.86	11.42	-	-	-	-	-	-	-	-
	Full fine-tuning	13016M	12.47	31.92	11.37	21.63	23.66	31.77	33.45	23.75	-	-	-
	CorDA	501M	22.77	58.76	9.99	17.38	17.93	30.70	31.29	26.97	0.93	0.87	0.90
	PiSSA	501M	18.03	56.60	9.60	19.41	20.70	32.01	33.45	27.11	0.84	0.94	0.89
	MiLoRA	501M	22.22	59.95	8.81	17.93	18.85	29.98	31.18	26.99	0.90	0.88	0.89
	LoRA	501M	20.69	60.38	8.56	17.93	19.22	30.10	32.01	26.98	0.87	0.89	0.88
	LoRA-Null	501M	23.61	59.93	8.37	17.01	18.48	29.62	31.06	26.87	0.91	0.86	0.88
	KaSA	501M	23.53	**61.21**	**10.97**	17.30	18.33	29.65	30.93	27.42	**0.99**	0.86	0.92
	FARSS (R=128)	400M	**23.85**	58.89	9.94	18.85	19.59	31.65	32.97	27.96	0.95	0.92	0.93
	FARSS (R=155)	501M	22.80	58.98	10.19	**20.89**	**22.74**	**33.81**	**35.49**	**29.27**	0.94	**1.01**	**0.98**
Gemma-2-9B	origin	-	26.12	68.02	18.95	-	-	-	-	-	-	-	-
	Full fine-tuning	9242M	20.33	53.16	19.29	34.20	36.78	44.24	46.76	36.39	-	-	-
	CorDA	432M	24.46	66.33	11.22	16.08	17.01	29.02	29.86	27.71	0.83	0.56	0.67
	PiSSA	432M	25.62	66.21	25.59	32.16	33.83	41.25	43.29	38.28	1.10	0.93	1.01
	MiLoRA	432M	27.04	68.36	24.70	20.89	22.37	29.62	30.94	31.99	**1.11**	0.64	0.81
	LoRA	432M	**27.15**	**68.39**	23.87	25.32	26.80	34.41	35.85	34.54	1.10	0.75	0.89
	LoRA-Null	432M	26.68	68.36	23.18	22.18	23.66	33.57	35.13	33.25	1.08	0.70	0.85
	KaSA	432M	24.71	66.44	17.86	20.15	21.07	28.54	29.38	29.74	0.96	0.61	0.74
	FARSS (R=128)	363M	25.10	66.18	25.44	33.09	35.12	**43.41**	**45.80**	39.16	1.09	0.97	**1.03**
	FARSS (R=155)	432M	25.32	65.84	**26.03**	**35.30**	**37.15**	43.17	45.08	**39.70**	1.10	**1.00**	**1.03**

Table 9: Knowledge-preserved adaptation results on Instruction Following (IFEval) with world knowledge scores (TriviaQA, NQ-open, WebQS) and summary metrics K, T, and KT.