Beyond Structural Agnosticism: Stable-Rank-Guided LoRA for Structure-Aware Fine-Tuning

Yuanyang Cao[1], **Xichun Liu**[1], **Fuwei Zhang**[2], **Meiqin Liu**[3], **Jianji Wang**[1*]

[1]State Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi, China

[2]Institute of Artificial Intelligence, Beihang University, Beijing, China

[3]College of Electrical Engineering, Zhejiang University, Hangzhou, Zhejiang, China

caoyuan1418@stu.xjtu.edu.cn, liuxichun@stu.xjtu.edu.cn
zhangfuwei@buaa.edu.cn, liumeiqin@zju.edu.cn
wangjianji@mail.xjtu.edu.cn

Abstract

Current parameter-efficient fine-tuning (PEFT) methods like Low-Rank Adaptation (LoRA) are structurally agnostic, applying uniform configurations across all layers, which overlooks their vast functional heterogeneity. We propose Structure-Aware LoRA (SA-LoRA), a new framework that automatically tailors fine-tuning intensity to each layer's intrinsic complexity. Specifically, we leverage the Stable Rank as a spectral metric to align the adaptation magnitude of each layer with its pre-trained spectral structure, enabling an automated, principled allocation of learning capacity that directly addresses the structural agnosticism of existing methods. To enhance adaptability and robustness, we introduce a hybrid calibration mechanism that fuses the task-agnostic prior with task-specific gradient feedback, underpinned by a budget-conservation principle to ensure stability. Extensive experiments demonstrate that SA-LoRA consistently outperforms strong PEFT baselines, often achieving state-of-the-art performance with enhanced stability. The code is available at `https://github.com/E-green11/SA-LORA`.

1 Introduction

The advent of large-scale pre-trained models, particularly Large Language Models (LLMs) [3], has fundamentally reshaped the landscape of machine learning. The dominant paradigm for leveraging these powerful models involves adapting them to specific downstream tasks through a process known as fine-tuning [9]. However, as model sizes scale to hundreds of billions of parameters [5], full fine-tuning has become computationally prohibitive, requiring extensive GPU resources and memory far beyond the reach of most researchers and practitioners [1]. This challenge has catalyzed the development of parameter-efficient fine-tuning (PEFT) methods [32, 34], which seek to match the performance of full fine-tuning while updating only a small fraction of the model's parameters [17, 37, 38]. Among these, Low-Rank Adaptation (LoRA) [19] has emerged as an effective and widely adopted technique. LoRA operates by freezing the original pre-trained weights and injecting small, trainable low-rank matrices into the network's layers, which drastically reduces the number of trainable parameters and the associated memory footprint.

Despite their success, a key limitation of current PEFT methods is that they remain largely structure-agnostic [27]. They often rely on globally shared fine-tuning rules, such as a single LoRA rank or

*Corresponding author.

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

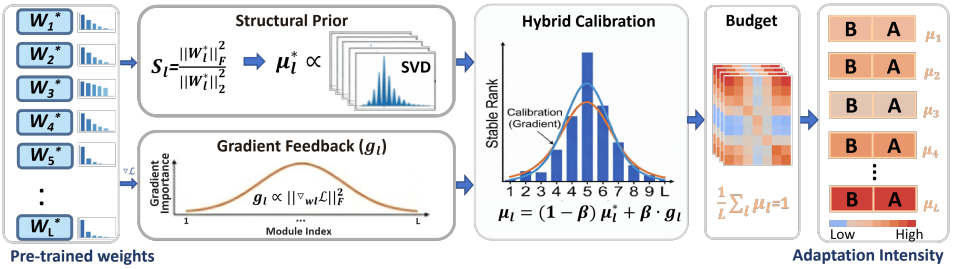

Figure 1: Overview of the SA-LoRA framework. The process begins by analyzing the pre-trained weights to compute the stable rank, which quantifies the structural complexity of each layer. The structural prior is then integrated with dynamic gradient feedback via a hybrid calibration mechanism to yield layer-specific multipliers μ_l. The multipliers adjust the adaptation intensity for each layer's LoRA parameters, facilitating structure-aware fine-tuning while maintaining budget conservation.

learning-rate scheme, across layers of a deep neural network. The one-size-fits-all strategy overlooks the vast functional and structural heterogeneity in pre-trained models. For instance, the shallow layers responsible for surface-level feature extraction are architecturally and functionally distinct from the deep layers that handle complex semantic reasoning [22, 56]. Although some recent works address this heterogeneity by proposing asymmetric learning rates for LoRA matrices [16], these strategies are still uniformly applied across all layers, thus failing to solve the core issue of inter-layer differences. We posit that this pervasive structural agnosticism is a fundamental and still underexplored source of suboptimality in modern PEFT practice, especially for large-scale task adaptation.

In this paper, we argue that the fine-tuning strategy for different layers should not be simply grounded in a global search but rather be intrinsically grounded in the mathematical properties of each layer's weights. To instantiate this principle, we introduce Structure-Aware LoRA (SA-LoRA), which is a novel PEFT framework that dynamically allocates adaptation capacity across layers via structure-aware multipliers. As shown in Figure 1, our core methodology leverages spectral analysis to quantify layer-wise complexity. Specifically, we employ the stable rank [14] as a proxy for the weight matrix's information content, enabling a principled, automated resource allocation rooted in optimization theory. By capturing the effective dimensionality, the Stable Rank provides a theoretically motivated signal that dictates the adaptation magnitude for each layer. To further bolster the robustness and adaptability of our framework, we propose two pivotal mechanisms. First, we design a hybrid calibration mechanism that fuses our task-agnostic structural prior grounded in Stable Rank with task-specific dynamic feedback obtained from the gradient energy of the network. The mechanism allows the framework to finely align itself with the specific requirements of downstream tasks. Second, we enforce a budget conservation principle to ensure training stability. By using a global normalization constraint, we ensure that the total adaptation capacity of the model remains invariant, meaning that SA-LoRA acts as an intelligent redistribution mechanism for the learning budget rather than inflating memory consumption. To validate our proposed approach, through extensive experiments, we demonstrate that SA-LoRA consistently surpasses standard LoRA and its variants across diverse models and tasks, achieving superior performance while simultaneously enhancing training stability and mitigating the sensitivity to hyperparameter tuning.

In summary, our main contributions are threefold:

- **Structure-Aware PEFT:** We introduce SA-LoRA, a PEFT framework that addresses the structural agnosticism of existing methods by replacing uniform policies with an automated, layer-aware training allocation strategy.
- **Stable-Rank-Guided Allocation:** We propose a spectral adaptation mechanism that leverages stable rank to guide per-layer adaptation intensity in PEFT, providing a first-order connection between intrinsic spectral properties and fine-tuning dynamics.
- **Hybrid Calibration with Budget Control:** We introduce a hybrid calibration mechanism that combines structural priors with gradient feedback under a budget conservation principle, improving performance and training stability.

2 Related Work

Parameter-Efficient Fine-Tuning. The cost of full fine-tuning has driven the development of diverse PEFT methods, which are commonly grouped into additive, selective, and reparameterization-based families [10]. Additive methods, such as Adapters [18, 48, 51], insert small trainable modules into the pre-trained network. Selective methods, such as BitFit [66] and Diff Pruning [15], update only a small subset of the original parameters. Reparameterization-based methods have become particularly influential, with LoRA [19] decomposing weight updates into low-rank matrices ($\Delta W = BA$) and emerging as a standard approach for efficient LLM fine-tuning [55].

Subsequent work has improved LoRA along several axes. Efficiency-oriented methods such as QLoRA [8], VeRA [30], and LoftQ [35] reduce memory or parameter cost. Rank-adaptive methods such as AdaLoRA [69] and DyLoRA [58] redistribute rank budgets across layers. Architecture-oriented variants including DoRA [40] and rsLoRA [26] modify the update parameterization or scaling, while LoRA+ and ALLoRA [16, 20] focus on learning-rate design. Despite these advances, most methods still apply uniform adaptation policies across layers and do not explicitly exploit layer-wise structural heterogeneity. Our work addresses this gap by deriving a structural prior directly from the spectral properties of pre-trained weights and using it to guide layer-wise adaptation intensity.

Structural Heterogeneity in Neural Networks. A complementary line of work studies the spectral structure of weight matrices and its implications for layer-wise heterogeneity [45]. Singular value spectra vary across depth and serve as signatures of layer function [50, 65]. Stable rank, in particular, provides a robust proxy for effective dimensionality and information complexity [31]. It also reflects how information and curvature are distributed across directions: high stable rank indicates distributed structure [68], whereas low stable rank corresponds to concentration in dominant directions [47]. More broadly, spectral properties have been linked to generalization and optimization behavior [7, 53, 61], including how fine-tuning perturbations propagate through the network.

Despite these insights, PEFT remains largely structure-agnostic, treating heterogeneous layers as homogeneous entities. The link between pre-trained structure and adaptation allocation has not been systematically explored. We address this gap by using stable rank as a structural prior for layer-wise fine-tuning intensity, yielding a principled, layer-aware alternative to uniform LoRA policies.

3 Methodology

3.1 Preliminaries: Per-Layer LoRA Dynamics

Consider a single layer within a pre-trained Transformer, parameterized by a frozen weight matrix $W_l^* \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ (denoting specific modules like W_Q, W_V). LoRA adapts this layer by introducing a low-rank update $W_l^{\Delta} = B_l A_l$, where $B_l \in \mathbb{R}^{d_{\text{out}} \times r}$ and $A_l \in \mathbb{R}^{r \times d_{\text{in}}}$ are the trainable low-rank matrices (with rank $r \ll \min(d_{\text{in}}, d_{\text{out}})$). The effective forward pass is given by:

$$h_l(x) = (W_l^* + B_l A_l)x, \tag{1}$$

where A_l is initialized with random Gaussian noise and B_l with zeros, ensuring that the initial update is zero ($B_l A_l = 0$). A core challenge lies in setting appropriate learning rates, η_{A_l} and η_{B_l}, for these matrices. The gradients with respect to A_l and B_l are derived as follows:

$$\frac{\partial \mathcal{L}}{\partial A_l} = B_l^T G_l, \qquad \frac{\partial \mathcal{L}}{\partial B_l} = G_l A_l^T, \tag{2}$$

where $G_l \triangleq \frac{\partial \mathcal{L}}{\partial (B_l A_l)} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ represents the gradient with respect to the low-rank update.

3.2 Deriving Structure-Aware Adaptation from Scaling Laws

We ground our derivation in the distinction between feature learning [28] and the lazy regime established in infinite-width network theory, reviewed in Appendix A. The lazy regime occurs when parameter updates are negligible relative to initialization, effectively freezing pre-trained features. In contrast, feature learning requires updates significant enough to reshape representations. Standard scaling laws suggest calibrating adaptation intensity by the global network width n to ensure feature learning, but the rule assumes a uniform network structure and overlooks heterogeneity across layers.

We propose that these scaling laws should be localized to each layer and that the fine-tuning dynamics of layer l are not governed by the global nominal width n, but by a layer-specific effective width, $n_{\text{eff}}(l)$, which reflects the intrinsic complexity of the transformation. Therefore, we measure this effective complexity using the stable rank, a theoretically grounded metric that quantifies the effective dimensionality of the pre-trained weight matrix W_l^*. The stable rank of a matrix M is defined as:

$$\text{srank}(M) = \frac{\|M\|_F^2}{\|M\|_2^2} = \frac{\sum_i \sigma_i(M)^2}{\sigma_{\max}(M)^2}, \tag{3}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\|\cdot\|_2$ is the spectral norm, and $\sigma_i(M)$ are the singular values of M. The stable rank possesses several desirable properties. It is bounded by $1 \leq \text{srank}(M) \leq \text{rank}(M)$, where the lower bound corresponds to highly anisotropic transformations dominated by a single direction, while the upper bound signifies isotropic behavior across all dimensions. Importantly, the stable rank has an information-theoretic interpretation: letting $p_i = \sigma_i^2 / \|M\|_F^2$ define a probability distribution over singular directions (spectral density), we have:

$$\log(\text{srank}(M)) = H_{\min}(M) \leq H(M) = -\sum_i p_i \log p_i, \tag{4}$$

where $H_{\min}(M) = -\log(p_{\max})$ is the min-entropy and $H(M)$ is the Shannon spectral entropy. The relationship establishes the stable rank as a proxy for the effective number of degrees of freedom (exponential of entropy), characterizing the information capacity of the layer (justification in Appendix B). Furthermore, as formally shown in Appendix D, the stable rank is robust to small perturbations Δ when both $\|\Delta\|_2 \ll \|M\|_2$ and $\|\Delta\|_F \ll \|M\|_F$:

$$\text{srank}(M + \Delta) \approx \text{srank}(M). \tag{5}$$

The robustness ensures that $n_{\text{eff}}(l)$ remains a reliable measure of the layer's core complexity.

Following the localized scaling argument, we replace the global nominal width n with $n_{\text{eff}}(l)$. The substitution leads to our central layer-wise complexity principle: to ensure consistent feature learning across heterogeneous layers, the magnitude of the adaptation must be calibrated to the layer's intrinsic complexity. We formalize this by introducing a layer-specific adaptation multiplier μ_l:

$$\mu_l \propto n_{\text{eff}}(l) = \text{srank}(W_l^*), \tag{6}$$

where μ_l governs the adaptation intensity. We assign higher intensity to high-rank layers to counteract signal dilution: because representational capacity is distributed across many dimensions, small updates fail to perturb features meaningfully, trapping the layer in the lazy regime. To motivate this scaling beyond heuristics, we analyze the optimization geometry in Appendix E. This analysis should be interpreted as a geometric rationale for practical optimization efficiency rather than a full convergence-rate proof. Optimization theory dictates that the learning rate η is bounded by loss-landscape sharpness: $\eta \leq 2/\lambda_{\max}(\mathcal{H}_l)$. In our framework, stable rank is not assumed to exactly determine sharpness; instead, it serves as a first-order structural ordering signal under layer-dependent factors treated as same-order across comparable Transformer blocks within the same backbone. Under this conditional approximation, higher-stable-rank layers tend to correspond to flatter, less concentrated geometry, while lower-stable-rank layers remain more sensitive to aggressive updates. Under a fixed global budget, uniform LoRA must remain conservative enough for the most sensitive layers, whereas SA-LoRA redistributes the same budget toward layers that can make larger safe progress, providing an optimization rationale for faster empirical loss reduction. Conversely, low-rank layers concentrate power in few directions, allowing effective modification with smaller updates.

Formally, we quantify the geometric connection by modeling the local loss as $\Delta\mathcal{L} \approx \frac{1}{2}\text{Tr}((\Delta W_l)^T \mathcal{H}_l \Delta W_l)$, where $\mathcal{H}_l$ is the Hessian. Under this structural approximation, the curvature complexity of the loss landscape, measured by the ratio of total to peak sensitivity, is approximated by a layer-dependent factor times the stable rank of the weights:

$$\frac{\text{Tr}(\mathcal{H}_l)}{\|\mathcal{H}_l\|_2} \approx \gamma_l \cdot \frac{\|W_l^*\|_F^2}{\|W_l^*\|_2^2} = \gamma_l \, \text{srank}(W_l^*), \tag{7}$$

where γ_l is an unknown layer-dependent factor. Stable rank therefore captures a first-order component of each layer's local geometry, while hybrid calibration corrects residual mismatch beyond the static

prior. We next express the structural prior through the singular value distribution. Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$ be the singular values of W_l^*. The stable rank can be decomposed as:

$$\mathrm{srank}(W_l^*) = \frac{\sum_{i=1}^{r} \sigma_i^2}{\sigma_1^2} = 1 + \sum_{i=2}^{r} \left(\frac{\sigma_i}{\sigma_1}\right)^2 . \tag{8}$$

The decomposition reveals that the stable rank quantifies the spectral tail mass beyond the dominant singular value, providing a principled measure of how distributed the layer's information content is across its singular subspaces. Further justification is provided in Appendix E and D.

3.3 Structure-Aware Adaptation Allocation

To operationalize the principle in Eq. (6), we employ a per-layer multiplier, μ_l, to modulate the adaptation intensity. Rather than applying uniform updates, the mechanism dynamically allocates optimization capacity based on structural complexity. The allocation proceeds in three steps:

Step 1: Raw Score Computation. For each LoRA-targeted layer l, we compute its raw structural complexity score, s_l, as the stable rank of its frozen pre-trained weight matrix W_l^* defined in Eq. (3).

Step 2: Score-to-Multiplier Mapping. We convert the raw scores s_l into a set of preliminary multipliers m_l. Given the set of scores from all layers $\{s_k\}_{k=1}^{L}$, we first normalize each s_l to the range $[0, 1]$ via min-max normalization:

$$\bar{s}_l = \frac{s_l - \min_k(s_k)}{\max_k(s_k) - \min_k(s_k) + \epsilon} , \tag{9}$$

where ϵ is added for numerical stability. Then, these normalized scores are mapped to a target range $[\mu_{\min}, \mu_{\max}]$ through a power transformation with exponent p:

$$m_l = \mu_{\min} + (\mu_{\max} - \mu_{\min}) \cdot (\bar{s}_l)^p . \tag{10}$$

The exponent p serves as a modulation factor: $p > 1$ concentrates adaptation capacity on high-complexity layers, whereas $p < 1$ encourages a uniform distribution of resources across layers.

Step 3: Budget-Conserving Normalization. To ensure training stability, we enforce a strict budget-conservation constraint. We normalize the preliminary multipliers m_l such that their mean is 1.0:

$$\mu_l = \frac{m_l}{\frac{1}{L}\sum_{k=1}^{L} m_k}, \qquad \frac{1}{L}\sum_{l=1}^{L} \mu_l = 1 . \tag{11}$$

This normalization ensures that SA-LoRA operates as a fixed-budget redistribution mechanism: it amplifies high-complexity layers at the expense of simpler ones without inflating the global optimization budget. Importantly, budget conservation should be interpreted as a safety and feasibility constraint, not as a guarantee that the resulting multipliers are correctly ranked or perfectly spaced relative to the unknown layer-wise optimum. Its role is to bound the global update scale and prevent pathological over-allocation when the structural prior is imperfect. The layer-wise correction is instead provided by the hybrid gradient calibration in Section 3.4. The multiplier μ_l modulates the per-layer optimization behavior by scaling the optimizer's step size and applies to each target module.

3.4 Hybrid Calibration and Online Evolution

To enhance adaptability, we calibrate the structural prior using task-specific gradients. Our theoretical foundation links gradient statistics to the Fisher Information Matrix (FIM), which quantifies parameter sensitivity [46]. For parameters θ_l, the FIM is defined as:

$$\mathcal{I}(\theta_l) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\nabla_{\theta_l} \log p(y|x,\theta) \cdot \nabla_{\theta_l} \log p(y|x,\theta)^\top \right] . \tag{12}$$

The FIM captures the local curvature of the KL-divergence between the model's predictive distribution and small perturbations thereof. Its trace $\mathrm{tr}(\mathcal{I}(\theta_l)) = \sum_i \lambda_i(\mathcal{I}(\theta_l))$ quantifies the total task sensitivity of layer l, where λ_i are the eigenvalues of the FIM. A key insight, further discussed in Appendix C and Appendix F, is that the expected squared gradient norm over the adapted parameters matches

this trace via the Fisher identity, while the gradient energy used in our implementation serves as a practical proxy for this quantity:

$$\mathbb{E}\left[\|\nabla_{\theta_l}\mathcal{L}\|_F^2\right] = \mathrm{tr}\left(\mathbb{E}\left[\nabla_{\theta_l}\mathcal{L}\cdot\nabla_{\theta_l}\mathcal{L}^\top\right]\right) = \mathrm{tr}(\mathcal{I}(\theta_l)). \tag{13}$$

Initial Gradient Calibration. We operationalize this insight by refining the static multipliers using gradient statistics. For notation simplicity, we express this proxy in terms of the virtual low-rank update $\Delta W_l = B_l A_l$. In implementation, however, we do not explicitly materialize dense gradients for the corresponding virtual dense update; instead, the same layer-wise energy is estimated in a memory-efficient manner from the available LoRA gradients. During the first N_{calib} steps, we accumulate the gradient energy for each layer l:

$$E_l = \sum_{t=1}^{N_{\text{calib}}} \left\| \frac{\partial \mathcal{L}_t}{\partial \Delta W_l^{(t)}} \right\|_F^2 \approx N_{\text{calib}} \cdot \mathrm{tr}(\mathcal{I}(\theta_l)). \tag{14}$$

To align with the structural multipliers, we convert the raw energy E_l into gradient scores g_l via mean-normalization ($g_l = E_l/\bar{E}$). We then fuse them with the structural prior to update the multiplier:

$$\mu_l \leftarrow (1-\beta_{\text{blend}})\cdot \mu_l^{\text{struct}} + \beta_{\text{blend}} \cdot g_l, \tag{15}$$

where $\beta_{\text{blend}} \in [0,1]$ controls the fusion weight. Following the fusion, we reapply budget conservation to the updated set $\{\mu_l\}$ to maintain global stability.

Online Adaptation Evolution. For real-time adaptability, we track energy via an Exponential Moving Average (EMA) with decay rate β_{ema}:

$$E_l^{(t)} = \beta_{\text{ema}} \cdot E_l^{(t-1)} + (1-\beta_{\text{ema}}) \cdot \left\| \frac{\partial \mathcal{L}_t}{\partial \Delta W_l^{(t)}} \right\|_F^2. \tag{16}$$

Unrolling this recursion reveals the effective window:

$$E_l^{(t)} = (1-\beta_{\text{ema}}) \sum_{\tau=0}^{t-1} \beta_{\text{ema}}^{\tau} \left\| \frac{\partial \mathcal{L}_{t-\tau}}{\partial \Delta W_l^{(t-\tau)}} \right\|_F^2, \tag{17}$$

where the effective window size is approximately $1/(1-\beta_{\text{ema}})$. This formulation shows that recent gradients receive exponentially higher weight, allowing adaptation to non-stationary optimization dynamics while maintaining smoothness. The stability of this online evolution process is theoretically analyzed in Appendix G. The complete process of the SA-Lora is detailed in Algorithm 1.

Algorithm 1 The SA-LoRA Framework

Require: Pre-trained weights $\{W_l^*\}$; Base rate η; Hyperparams $p, \beta_{\text{blend}}, N_{\text{calib}}, \beta_{\text{ema}}, k_{\text{update}}$

Initialize LoRA $\{A_l, B_l\}$; Optimizer $\mathcal{O}$; $E_l \leftarrow 0$ for all l
// Phase 1: Structural Complexity Analysis
for $l = 1$ to L **do**
  $s_l \leftarrow \mathrm{srank}(W_l^*)$
end for
Compute $\{\mu_l^{\text{struct}}\}$ from $\{s_l\}$ via Steps 2 & 3
Initialize $\mu_l \leftarrow \mu_l^{\text{struct}}$ and update optimizer
// Phase 2: Training with Adaptive Calibration
for $t = 1$ to N_{steps} **do**
  Compute loss $\mathcal{L}_t$ and gradients $\nabla\mathcal{L}_t$
  if $t \le N_{\text{calib}}$ **then**
    Estimate $E_l \leftarrow E_l + \|\nabla_{\Delta W_l}\mathcal{L}_t\|_F^2$ from LoRA gradients for all l
    if $t = N_{\text{calib}}$ **then**
      Compute scores $\{g_l\}$; Blend & Re-normalize $\{\mu_l\}$; Update optimizer
    end if
  else
    // Online Evolution: Track EMA every step
    $E_l \leftarrow \beta_{\text{ema}} E_l + (1-\beta_{\text{ema}})\cdot$ estimated $\|\nabla_{\Delta W_l}\mathcal{L}_t\|_F^2$
    if $t \mod k_{\text{update}} = 0$ **then** Recompute $\{\mu_l\}$; Update optimizer
  end if
  $\mathcal{O}$.step()
end for

4 Experiments

4.1 Experimental Setup

Models and Tasks. We conduct comprehensive experiments across four evaluation scenarios. (1) Natural Language Understanding: We fine-tune RoBERTa-base (125M) and RoBERTa-large (355M) [41] on the GLUE benchmark [59], covering eight tasks including MNLI, QQP, SST-2, QNLI, CoLA, RTE, MRPC, and STS-B. (2) Natural Language Generation: We evaluate GPT-2 Medium (355M) and GPT-2 Large (774M) [49] on the E2E NLG Challenge [11], reporting BLEU, NIST, METEOR, ROUGE-L, and CIDEr metrics. (3) Instruction Following: We fine-tune Llama-7B/13B and Llama2-7B/13B [57] and evaluate on MT-Bench with GPT-4 as the judge. (4) Multimodal Evaluation: We evaluate a VL-BART backbone on VQAv2, GQA, NLVR2, and MSCOCO Captioning.

Baselines. We compare SA-LoRA against a broad set of PEFT baselines: (1) LoRA [19] and Adapters [18, 48]; (2) LoRA+ [16]; (3) DoRA [40]; (4) BitFit [66]; (5) VeRA [30]; (6) DyLoRA [58], AdaLoRA [69], LoRA-FA [67] and StelLA [36].

Implementation Details. All experiments are conducted on NVIDIA A800 80GB GPUs. Detailed settings and dataset descriptions are provided in Appendix H.

4.2 Main Results

We compare SA-LoRA against strong PEFT baselines across four evaluation scenarios. Unless otherwise noted, the reported results are averaged over three independent runs. The results consistently support the effectiveness of our structure-aware adaptation strategy.

Natural Language Understanding. As shown in Table 1, SA-LoRA achieves top-tier performance on the GLUE benchmark compared to existing PEFT methods. For RoBERTa-base, SA-LoRA obtains an average GLUE score of 88.0, outperforming LoRA (87.2) and other adapter-based methods. Notably, SA-LoRA achieves substantial gains on SST-2 (95.7%) and QQP (92.0%), demonstrating the benefit of layer-aware adaptation for both sentiment analysis and paraphrase detection tasks. For RoBERTa-large, SA-LoRA remains competitive with 0.8M trainable parameters, surpassing methods with larger parameter budgets.

Natural Language Generation. As shown in Table 2, we evaluate SA-LoRA's effectiveness for generation tasks on the E2E NLG benchmark using GPT-2 Medium and Large. SA-LoRA achieves the strongest overall results across BLEU, NIST, METEOR, R-L, and CIDEr compared to the LoRA, LoRA+, DyLoRA, AdaLoRA, and VeRA baselines. The results indicate that our structure-aware multiplier generalizes well beyond classification tasks to conditional text generation, where different layers contribute differently to content fidelity.

Instruction Following. As shown in Table 3, we further evaluate SA-LoRA on the MT-Bench benchmark using Llama-7B/13B and Llama2-7B/13B, with GPT-4 as the judge. This challenging benchmark tests instruction-following capabilities across multiple turns of conversation. SA-LoRA performs strongly and demonstrates a 0.45 improvement over baseline LoRA on Llama-7B, validating that our principled adaptation allocation scales effectively to large language models and complex reasoning tasks. A supplementary convergence comparison on the Llama-7B setting is provided in Appendix J. Qualitative case studies with detailed response comparisons are provided in Appendix P.

Multimodal Evaluation. Table 4 reports a supplementary vision-language evaluation on a VL-BART backbone with CLIP-ResNet101 and BART. Under a matched trainable ratio, SA-LoRA improves over LoRA on all four multimodal tasks and outperforms DoRA on VQAv2, GQA, NLVR2, and COCO Captioning. These results provide initial evidence that the combination of a spectral structural prior and task-driven calibration remains effective beyond pure language settings.

Across the evaluation scenarios above, SA-LoRA consistently demonstrates the advantage of structure-aware adaptation. The results support our central hypothesis that tailoring adaptation intensity to layer complexity, as quantified by stable rank, is more effective than applying uniform policies. With the main global hyperparameters held fixed, the observed gains are consistent with the contribution

Table 1: Performance comparison of various PEFT methods on the GLUE benchmark. We report the evaluation metric (Matthews correlation for CoLA, Pearson correlation for STS-B, and Accuracy for other tasks) for each task. Results are reported as mean$_{\pm \text{std}}$ across three random seeds. **Bold** indicates the best result, and underlined indicates the second best.

	Method	#Params	MNLI	QQP	SST-2	QNLI	CoLA	RTE	MRPC	STSB	Avg.
RoBERTa-base	Full FT	125.0M	$87.6_{\pm.2}$	$91.9_{\pm.3}$	$94.8_{\pm.4}$	$92.8_{\pm.6}$	$63.6_{\pm.9}$	$78.7_{\pm.8}$	$90.2_{\pm.5}$	$91.2_{\pm1.2}$	86.4
	BitFit	0.1M	$84.7_{\pm.2}$	$84.0_{\pm.1}$	$93.7_{\pm.3}$	$91.8_{\pm.1}$	$62.0_{\pm.2}$	$81.5_{\pm.3}$	**92.7**$_{\pm.1}$	$90.8_{\pm.3}$	85.2
	Adpt^P	0.3M	$87.1_{\pm.1}$	$90.2_{\pm.1}$	$94.2_{\pm.1}$	$93.1_{\pm.1}$	$60.8_{\pm.4}$	$71.5_{\pm2.7}$	$88.5_{\pm1.1}$	$89.7_{\pm.3}$	84.4
	Adpt^H	0.3M	$86.6_{\pm.2}$	$89.9_{\pm.2}$	$94.0_{\pm.3}$	$92.8_{\pm.2}$	$60.4_{\pm.7}$	$72.1_{\pm1.9}$	$88.3_{\pm.8}$	$89.0_{\pm.4}$	84.1
	LoRA	0.3M	$87.5_{\pm.3}$	$90.8_{\pm.1}$	$95.1_{\pm.2}$	$93.3_{\pm.3}$	$63.4_{\pm1.2}$	**86.6**$_{\pm.7}$	$89.7_{\pm.7}$	$91.5_{\pm.2}$	87.2
	DyLoRA	0.3M	$87.6_{\pm.2}$	$90.9_{\pm.2}$	$94.1_{\pm.2}$	$92.2_{\pm.1}$	$61.2_{\pm1.0}$	$71.0_{\pm1.2}$	$88.5_{\pm.8}$	$89.6_{\pm.3}$	84.4
	AdaLoRA	0.3M	$87.7_{\pm.1}$	$91.0_{\pm.2}$	$95.3_{\pm.2}$	$93.3_{\pm.3}$	$64.8_{\pm.9}$	$85.5_{\pm.9}$	$90.8_{\pm.6}$	$91.6_{\pm.1}$	87.5
	LoRA-FA	1.4M	$87.3_{\pm.3}$	$91.0_{\pm.2}$	$95.0_{\pm.2}$	$93.1_{\pm.2}$	$64.9_{\pm.9}$	$83.7_{\pm1.1}$	$90.9_{\pm.6}$	$90.9_{\pm.5}$	87.1
	DoRA	0.3M	$88.3_{\pm.3}$	$91.6_{\pm.1}$	**95.7**$_{\pm.2}$	$93.0_{\pm.1}$	**66.6**$_{\pm1.2}$	$82.9_{\pm2.4}$	$92.4_{\pm.5}$	$91.3_{\pm.4}$	87.7
	VeRA	0.04M	$86.8_{\pm.3}$	$90.6_{\pm.2}$	$94.6_{\pm.1}$	$91.8_{\pm.2}$	$65.6_{\pm.8}$	$78.7_{\pm.7}$	$89.5_{\pm.5}$	$90.7_{\pm.2}$	86.0
	LoRA+	0.3M	$87.0_{\pm.1}$	$91.8_{\pm.1}$	$94.8_{\pm.2}$	$93.1_{\pm.3}$	$66.3_{\pm.6}$	$82.8_{\pm.8}$	$91.1_{\pm.4}$	$91.5_{\pm.4}$	87.3
	SA-LoRA	0.3M	**88.4**$_{\pm.2}$	**92.0**$_{\pm.1}$	**95.7**$_{\pm.3}$	**93.4**$_{\pm.1}$	$66.5_{\pm.8}$	$84.6_{\pm.8}$	$92.0_{\pm.2}$	**92.1**$_{\pm.1}$	**88.0**
RoBERTa-large	Full FT	355.0M	$90.2_{\pm2.6}$	$92.2_{\pm1.8}$	$96.4_{\pm2.8}$	$94.7_{\pm1.5}$	$68.0_{\pm1.1}$	**86.6**$_{\pm2.1}$	**90.9**$_{\pm.6}$	$92.4_{\pm1.8}$	88.9
	BitFit	0.2M	$89.9_{\pm.4}$	$91.1_{\pm.4}$	$96.2_{\pm.3}$	$94.3_{\pm.3}$	$66.5_{\pm1.1}$	$81.2_{\pm1.7}$	$90.5_{\pm.9}$	$92.7_{\pm.6}$	87.8
	Adpt^P	0.8M	$90.5_{\pm.3}$	$91.7_{\pm.2}$	$96.6_{\pm.2}$	$94.8_{\pm.3}$	$67.8_{\pm2.5}$	$80.1_{\pm2.9}$	$89.7_{\pm1.2}$	$91.9_{\pm.4}$	87.9
	Adpt^H	0.8M	$90.3_{\pm.3}$	$91.5_{\pm.1}$	$96.3_{\pm.5}$	$94.7_{\pm.2}$	$66.3_{\pm2.0}$	$72.9_{\pm2.9}$	$87.7_{\pm1.7}$	$91.5_{\pm.5}$	86.4
	LoRA	0.8M	$90.6_{\pm.2}$	$91.6_{\pm.1}$	$96.2_{\pm.5}$	$94.8_{\pm.3}$	$68.2_{\pm1.9}$	$85.2_{\pm1.1}$	$90.2_{\pm1.0}$	$92.3_{\pm.5}$	88.6
	DyLoRA	0.8M	$90.3_{\pm.4}$	$91.2_{\pm.3}$	$96.2_{\pm.3}$	$94.5_{\pm.2}$	$68.0_{\pm1.5}$	$80.0_{\pm1.2}$	$90.1_{\pm.9}$	$91.3_{\pm.6}$	87.7
	AdaLoRA	0.8M	$90.8_{\pm.3}$	$91.8_{\pm.5}$	$96.3_{\pm.2}$	$94.7_{\pm.4}$	$68.4_{\pm1.2}$	$85.8_{\pm1.0}$	$90.6_{\pm.7}$	$92.4_{\pm.5}$	88.9
	LoRA-FA	3.7M	$90.0_{\pm0.9}$	$91.2_{\pm0.8}$	$96.0_{\pm1.1}$	$94.4_{\pm0.5}$	$68.0_{\pm1.3}$	$86.1_{\pm1.1}$	$90.0_{\pm1.5}$	$92.0_{\pm0.9}$	88.4
	DoRA	0.8M	$90.8_{\pm.6}$	$92.4_{\pm.2}$	$96.6_{\pm.4}$	$94.7_{\pm.3}$	**68.5**$_{\pm3.4}$	$85.4_{\pm1.8}$	$90.7_{\pm3.2}$	$92.0_{\pm.9}$	88.9
	VeRA	0.1M	$90.0_{\pm.4}$	$91.0_{\pm.7}$	$96.1_{\pm.1}$	$94.4_{\pm.2}$	$68.0_{\pm.8}$	$85.9_{\pm.7}$	**90.9**$_{\pm.7}$	$91.7_{\pm.8}$	88.4
	LoRA+	0.8M	$90.2_{\pm.2}$	$91.3_{\pm.5}$	$96.1_{\pm.2}$	$94.1_{\pm.3}$	$68.3_{\pm1.6}$	$86.3_{\pm.8}$	$90.3_{\pm.5}$	$92.3_{\pm.7}$	88.7
	SA-LoRA	0.8M	**91.0**$_{\pm.3}$	**92.5**$_{\pm.1}$	**96.8**$_{\pm.3}$	**95.1**$_{\pm.1}$	$68.1_{\pm.5}$	$86.5_{\pm1.0}$	$90.5_{\pm1.2}$	**93.0**$_{\pm.2}$	**89.2**

Table 2: Results on the E2E NLG benchmark with GPT-2 models.

	Method	Params	BLEU	NIST	MET.	R-L	CIDEr
Medium	FT	354.9M	68.2	8.62	46.2	71.0	2.47
	Adpt^L	0.4M	66.3	8.41	45.0	69.8	2.40
	Adpt^H	11.1M	67.3	8.50	46.0	70.7	2.44
	DyLoRA	0.4M	69.2	8.75	46.3	70.8	2.46
	AdaLoRA	0.4M	68.2	8.58	44.1	70.7	2.35
	LoRA	0.4M	68.9	8.69	46.4	71.3	2.51
	LoRA+	0.4M	69.9	8.84	46.6	71.5	2.51
	StelLA	0.4M	70.2	8.82	46.7	71.7	2.52
	VeRA	0.1M	70.1	8.81	46.6	71.5	2.50
	SA-LoRA	0.4M	**70.5**	**8.89**	**46.9**	**72.0**	**2.53**
Large	FT	774.0M	68.5	8.78	46.0	69.9	2.45
	Adpt^L	0.9M	69.1	8.68	46.3	71.4	2.49
	Adpt^H	23.0M	68.9	8.70	46.1	71.3	2.45
	DyLoRA	0.8M	70.2	8.83	46.8	71.8	2.53
	AdaLoRA	0.8M	69.8	8.79	46.5	71.5	2.50
	LoRA	0.8M	70.1	8.80	46.7	71.9	2.52
	LoRA+	0.8M	70.4	8.87	46.8	72.1	2.53
	VeRA	0.2M	70.3	8.85	46.9	71.6	2.54
	StelLA	0.8M	70.6	8.89	47.0	71.9	2.55
	SA-LoRA	0.8M	**70.9**	**8.95**	**47.4**	**72.6**	**2.59**

Table 3: Instruction-following capabilities on MT-Bench, with scores provided by GPT-4.

Model	Method	Params	Score
Llama 7B	LoRA	159.9M	5.03
	LoRA+	159.9M	5.28
	VeRA	1.6M	4.77
	DoRA	161.3M	5.46
	SA-LoRA	159.9M	**5.48**
Llama 13B	LoRA	250.3M	5.31
	LoRA+	250.3M	5.52
	VeRA	2.4M	5.22
	DoRA	252.5M	5.61
	SA-LoRA	250.3M	**5.70**
Llama2 7B	LoRA	159.9M	5.19
	LoRA+	159.9M	5.74
	VeRA	1.6M	5.08
	DoRA	161.3M	5.95
	SA-LoRA	159.9M	**6.01**
Llama2 13B	LoRA	250.3M	5.77
	LoRA+	250.3M	5.96
	VeRA	2.4M	5.93
	DoRA	252.5M	6.04
	SA-LoRA	250.3M	**6.10**

of our structure-aware multiplier μ_l. Dataset-level significance checks against standard LoRA are reported in Appendix L, and detailed training cost analysis is provided in Appendix I.

4.3 Ablation Studies and Analysis

We conduct a series of ablation studies on RoBERTa-base across SST-2, MRPC, and STS-B tasks. We report the average score in the main text; per-task results are provided in Appendix K.

Table 4: Supplementary multimodal results on a VL-BART backbone (CLIP-ResNet101 + BART).

METHOD	# PARAMS (%)	VQAV2	GQA	NLVR2	COCO CAP
FT	100	66.9	56.7	73.7	112.0
LORA	5.93	65.2	53.6	71.9	115.3
LORA+	5.93	65.5	54.2	72.5	115.7
DORA	5.96	65.8	54.7	73.1	115.9
SA-LORA (OURS)	5.93	**66.4**	**55.6**	**73.9**	**116.7**

Table 5: Ablation study of different components on RoBERTa-base. **Struct**: Stable rank-based multiplier; **Budget**: Budget-conserving normalization; **Grad**: Hybrid gradient calibration.

METHOD	STRUCT	BUDGET	GRAD.	AVG.
LORA (BASELINE)				92.1
SA-LORA (STRUCT)	✓			92.9
SA-LORA (STATIC)	✓	✓		93.1
SA-LORA (FULL)	✓	✓	✓	**93.3**

Table 6: Ablation studies on RoBERTa-base: (a) mapping power p (0.5: concave, 1.0: linear, 2.5: ours); (b) hybrid calibration ($\beta = 0$: structural, $\beta = 1$: gradient, $\beta = 0.2$: hybrid).

(A) MAPPING POWER p		(B) HYBRID CALIBRATION	
POWER p	AVG.	STRATEGY	AVG.
0.5	92.5	$\beta = 0$	93.1
1.0	92.9	$\beta = 1$	92.3
2.5 **(OURS)**	**93.3**	$\beta = 0.2$	**93.3**

Component Analysis. Table 5 presents a progressive ablation study on RoBERTa-base to isolate module contributions. The structure-aware mechanism alone improves upon the LoRA baseline (92.1% to 92.9%), showing the spectral prior as the main performance driver. Budget-conserving normalization further increases performance to 93.1%, and adding hybrid gradient calibration raises the final score to 93.3%, indicating that task-specific dynamics improve the static structural allocation. To test the direction and specificity of the stable-rank prior, Appendix K.1 reports an inverted-allocation ablation, and Appendix K.2 compares stable rank against alternative redistribution signals.

Impact of Non-linear Mapping Power (p). We map the normalized scores using a power transformation with exponent p. Our default model uses $p = 2.5$, which creates a convex mapping that amplifies the differences between high- and low-complexity layers. As shown in Table 6(a), we compare this choice against a linear mapping ($p = 1.0$) and a concave mapping ($p = 0.5$). The superior performance of $p = 2.5$ validates our hypothesis that a non-linear allocation, which prioritizes high-complexity layers, is more effective than a uniform linear scaling.

Fusing Structural Priors with Gradient Feedback. We analyze the effect of our hybrid calibration mechanism by varying the fusion weight β_{blend}. As shown in Table 6(b), we compare three variants: one using only the static structural prior ($\beta_{\text{blend}} = 0.0$), one using only dynamic gradient feedback ($\beta_{\text{blend}} = 1.0$), and a hybrid model ($\beta_{\text{blend}} = 0.2$). The hybrid model yields the strongest trade-off between structural robustness and dynamic adaptability among the tested settings, confirming that combining task-agnostic structural priors with task-specific gradient feedback leads to synergistic improvement. To strengthen this comparison, Appendix K.3 further reports a well-tuned pure-gradient baseline with EMA smoothing and budget-conserving normalization.

Impact of LoRA Rank (r). We investigate whether SA-LoRA provides consistent gains across different parameter budgets by varying the LoRA rank $r \in \{4, 8, 16, 32, 64\}$. As shown in Figure 2, SA-LoRA consistently outperforms baseline LoRA across all rank configurations. The performance gap is more pronounced at lower ranks ($r = 4, 8$), suggesting that structure-aware adaptation is particularly beneficial under constrained parameter budgets. Full results on all GLUE tasks are provided in Appendix M.

Robustness to Multiplier Range. We examine SA-LoRA's robustness in Appendix N. While performance naturally varies with adaptation intensity, the selected default configuration remains broadly effective across tasks. Specifically, a consistent high-performance region exists across diverse tasks. This validates the practical robustness of our automated allocation strategy. Under the same RoBERTa-base SST-2, MRPC, and STS-B 3-seed protocol, we also report a module-scope ablation over broader target-module choices in Appendix K.4.

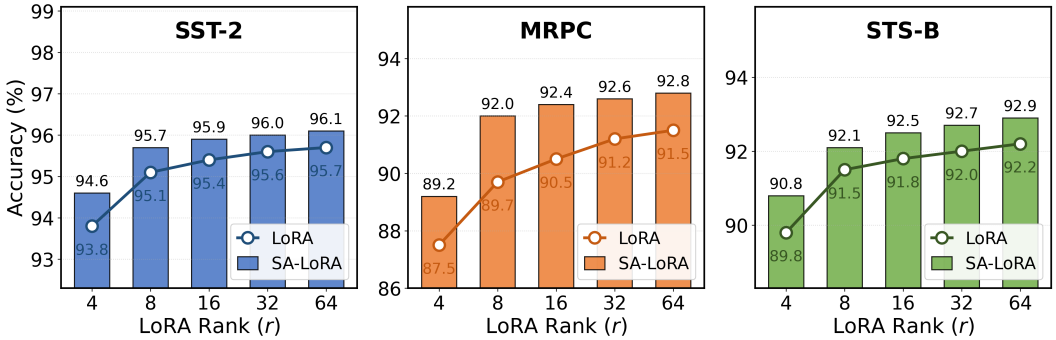

Figure 2: Impact of the rank on SA-LoRA performance.

4.4 Interpretability: Visualizing Allocation Dynamics

A key advantage of SA-LoRA is its interpretability. The final adaptation multipliers μ_l reveal task-dependent, layer-wise adaptation patterns. Appendix O visualizes these coefficients on RoBERTa-base for MNLI and SST-2, along with their layer-wise values. The results reveal alignment between allocated adaptation intensity and established findings on functional differences across Transformer layers, providing a view of where adaptation is emphasized. For MNLI, layers 6–12 exhibit elevated coefficients ($\mu_l > 1$), corresponding to middle-to-deep layers associated with syntactic and semantic abstractions. This suggests that complex linguistic relationships may benefit from stronger adaptation in these layers. By contrast, SST-2 yields a flatter distribution, suggesting that sentiment classification relies on more distributed adaptation, with less emphasis on individual layers.

Since the multipliers are normalized to a mean of one, these variations redistribute adaptation intensity without increasing the overall optimization budget. These contrasting patterns illustrate how the same pre-trained backbone can adjust its adaptation intensity to distinct task requirements. The stable-rank prior sets the structural preference, while gradient-based calibration refines it using task-specific optimization signals. Together, they yield interpretable layer-wise allocations without manually specified rules, facilitating comparisons across different tasks.

5 Conclusion and Limitations

In this work, we identify structural agnosticism as a core limitation of current PEFT methods and introduce SA-LoRA to address it through stable-rank-guided, layer-wise adaptation. The framework combines a structure-aware prior, hybrid gradient calibration, and budget-conserving normalization to support stable training. Across NLU, generation, instruction-following, and multimodal settings, SA-LoRA consistently outperforms standard LoRA and several strong variants while reducing sensitivity to global hyperparameter choices. Taken together, these results support stable rank as a useful first-order structural prior for practical PEFT allocation. More broadly, our results suggest that PEFT can benefit from treating layers as structurally heterogeneous rather than homogeneous.

Despite these promising results, SA-LoRA has several limitations. Its generalizability to broader architectures and tasks remains to be explored. Moreover, stable rank provides a structural prior rather than a guarantee of optimal layer-wise adaptation, while hybrid calibration remains sensitive to gradient signals and hyperparameters. Finally, our experiments primarily adopt fixed-rank LoRA configurations, leaving more flexible adaptation strategies unexplored. Future work will explore adaptive rank allocation, broader module selection, and more robust calibration mechanisms.

Acknowledgments and Disclosure of Funding

This work was supported in part by the New Generation Artificial Intelligence–National Science and Technology Major Project (No. 2025ZD0124202) and the National Natural Science Foundation of China under Grants 62495091, 62125305, and U23A20339. And the authors declare no competing interests. We sincerely thank the reviewers for their valuable feedback and constructive suggestions.

References

[1] 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.

[2] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *International conference on machine learning*, pages 254–263. PMLR, 2018.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Yuanyang Cao, Xichun Liu, Fuwei Zhang, Shangqi Deng, Ziyang Ren, and Jianji Wang. Beyond static allocation: Dynamic sensitivity-aware fine-tuning for vision transformers. In *Forty-third International Conference on Machine Learning*, 2026.

[5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113, 2023.

[6] Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. *arXiv preprint arXiv:2103.00065*, 2021.

[7] Amit Daniely and Eran Malach. Learning parities with neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 20356–20365, 2020.

[8] 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.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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*, pages 4171–4186, 2019.

[10] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*, 2022.

[11] Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156, 2020.

[12] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[13] Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 12799–12807, 2023.

[14] Tomer Galanti, Zachary S Siegel, Aparna Gupte, and Tomaso Poggio. Sgd and weight decay secretly minimize the rank of your neural network. *arXiv preprint arXiv:2206.05794*, 2022.

[15] Demi Guo, Alexander M Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pages 4884–4896, 2021.

[16] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: efficient low rank adaptation of large models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 17783–17806, 2024.

[17] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2022.

[18] 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 *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 2790–2799. PMLR, 2019.

[19] 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. *International Conference on Learning Representations (ICLR)*, 1(2):3, 2022.

[20] Hai Huang and Randall Balestriero. Allora: Adaptive learning rate mitigates lora fatal flaws. *arXiv preprint arXiv:2410.09692*, 2024.

[21] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[22] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[23] Dongyao Jiang, Haodong Jing, Yongqiang Ma, and Nanning Zheng. Beyond image classification: A video benchmark and dual-branch hybrid discrimination framework for compositional zero-shot learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9860–9869, 2025.

[24] Haodong Jing, Yongqiang Ma, Panqi Yang, Haibo Hua, and Nanning Zheng. Pinpointing visual content: Disentangled features in multimodal model for eeg representation learning and decoding. *Knowledge-Based Systems*, page 114212, 2025.

[25] Haodong Jing, Yongqiang Ma, Panqi Yang, Haoyu Li, Shuai Huang, Badong Chen, and Nanning Zheng. Damind: Zero-shot visual cross-domain alignment and representation for eeg decoding. *IEEE Transactions on Image Processing*, 2026.

[26] Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*, 2023.

[27] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*, volume 34, pages 1022–1035, 2021.

[28] Dhruva Karkada. The lazy (ntk) and rich (μp) regimes: a gentle tutorial. *arXiv preprint arXiv:2404.19719*, 2024.

[29] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[31] Antoine Ledent, Rodrigo Alves, and Yunwen Lei. Generalization bounds for rank-sparse neural networks. *arXiv preprint arXiv:2510.21945*, 2025.

[32] 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*. Association for Computational Linguistics, 2021.

[33] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

[34] 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)*. Association for Computational Linguistics, 2021.

[35] Y Li, Y Yu, C Liang, P He, N Karampatziakis, W Chen, and T Zhao. Loftq: Lora-fine-tuning-aware quantization for large language models. International Conference on Learning Representations, 2024.

[36] Zhizhong Li, Sina Sajadmanesh, Jingtao Li, and Lingjuan Lyu. Stella: Subspace learning in low-rank adaptation using stiefel manifold. *arXiv preprint arXiv:2510.01938*, 2025.

[37] Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*, 2023.

[38] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

[39] Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342*, 2023.

[40] 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 *Forty-first International Conference on Machine Learning*, 2024.

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

[42] Yuchen Lou, Zeqi Ye, and Minshuo Chen. Parameter-efficient subspace optimization for llm fine-tuning. *arXiv preprint arXiv:2512.02216*, 2025.

[43] Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of sgd in modern over-parametrized learning. In *International Conference on Machine Learning*, pages 3325–3334. PMLR, 2018.

[44] Charles H Martin and Michael W Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021.

[45] 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.

[46] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11264–11272, 2019.

[47] Samet Oymak, Zalan Fabian, Mingchen Li, and Mahdi Soltanolkotabi. Generalization guarantees for neural networks via harnessing the low-rank structure of the jacobian. *arXiv preprint arXiv:1906.05392*, 2019.

[48] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume*, pages 487–503, 2021.

[49] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[50] Noam Razin and Nadav Cohen. Implicit regularization in deep learning may not be explainable by norms. *Advances in neural information processing systems*, 33:21174–21187, 2020.

[51] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30, 2017.

[52] Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in neural information processing systems*, 33:20378–20389, 2020.

[53] Zhenmei Shi, Junyi Wei, and Yingyu Liang. A theoretical analysis on feature learning in neural networks: Emergence from inputs and advantage over fixed features. *arXiv preprint arXiv:2206.01717*, 2022.

[54] Sidak Pal Singh and Thomas Hofmann. Closed form of the hessian spectrum for some neural networks. In *High-dimensional Learning Dynamics 2024: The Emergence of Structure and Reasoning*, 2024.

[55] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. VL-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5227–5237, 2022.

[56] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

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

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

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

[60] Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. Svd-llm: Truncation-aware singular value decomposition for large language model compression. *arXiv preprint arXiv:2403.07378*, 2024.

[61] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[62] Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. *Advances in Neural Information Processing Systems*, 34:17084–17097, 2021.

[63] Panqi Yang, Haodong Jing, Nanning Zheng, and Yongqiang Ma. Unihoi: Unified human-object interaction understanding via unified token space. *arXiv preprint arXiv:2511.15046*, 2025.

[64] Yifan Yang, Jiajun Zhou, Ngai Wong, and Zheng Zhang. LoRETTA: Low-rank economic tensor-train adaptation for ultra-low-parameter fine-tuning of large language models. *arXiv preprint arXiv:2402.11417*, 2024.

[65] Zhihang Yuan, Yuzhang Shang, Yue Song, Dawei Yang, Qiang Wu, Yan Yan, and Guangyu Sun. ASVD: Activation-aware singular value decomposition for compressing large language models. *arXiv preprint arXiv:2312.05821*, 2023.

[66] 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.

[67] Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. LoRA-FA: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303*, 2023.

[68] Qingru Zhang, Simiao Zuo, Chen Liang, Alexander Bukharin, Pengcheng He, Weizhu Chen, and Tuo Zhao. Platon: Pruning large transformer models with upper confidence bound of weight importance. In *International conference on machine learning*, pages 26809–26823. PMLR, 2022.

[69] 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.

[70] Xiao Zhang, Haodong Jing, Yongqiang Ma, and Nanning Zheng. Decoupling primitive with experts: Dynamic feature alignment for compositional zero-shot learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

[71] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-Bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

A A Review of Learning Dynamics in the Infinite-Width Limit

To motivate our work, we first review the learning dynamics of LoRA in the infinite-width limit, drawing insights from recent analyses [16, 42]. This analysis is based on foundational principles like the Neural Tangent Kernel (NTK) regime [21] and Maximal Update Parametrization (μP) [62].

These theories show that for a network of width $n \to \infty$, preventing the learning dynamics from collapsing into the "lazy" NTK regime requires feature learning, where updates to the adapter matrices remain of a constant order, i.e., $\Delta A = \Theta(1)$ and $\Delta B = \Theta(1)$.

To derive the corresponding scaling of the learning rates, we analyze the scaling of the gradients. For this appendix, we follow an analysis-specific initialization convention used to expose the implied asymptotic scaling behavior, rather than to exactly mirror the practical LoRA initialization used in our implementation. Accordingly, this appendix should be interpreted as a motivating scaling-law analysis rather than a line-by-line derivation of the practical implementation. For theoretical analysis, we adopt an initialization scheme where $A_0 \approx 0$ and the $n \times r$ matrix B_0 contains entries of order $\Theta(1)$.

Scaling Analysis for A_l. The gradient for A_l is given by $\nabla_{A_l} = B_l^T \nabla_{W_l}$. In the μP framework, the gradient with respect to the pre-trained weights, ∇_{W_l}, remains order-stable under the adopted scaling convention. Since B_l^T is $r \times n$ with $\Theta(1)$ entries, the matrix product $B_l^T \nabla_{W_l}$ aggregates contributions across the width dimension n. Under this analysis-specific scaling view, the effective gradient scale for A_l grows linearly with width:

$$\|\nabla_{A_l}\|_F \asymp \left\|B_l^T \nabla_{W_l}\right\|_F = \Theta(n). \tag{18}$$

To achieve our goal of a constant-order update $\Delta A_l = \Theta(1)$, we must set the learning rate η_{A_l} to counteract this scaling:

$$\Delta A_l = \eta_{A_l} \cdot \nabla_{A_l} \implies \Theta(1) = \eta_{A_l} \cdot \Theta(n) \implies \eta_{A_l} = \Theta(1/n). \tag{19}$$

Scaling Analysis for B_l. The gradient for B_l is $\nabla_{B_l} = \nabla_{W_l} A_l^T$. At initialization, $\nabla_{B_0} \approx 0$ because $A_0 \approx 0$. However, after the first update step, $A_1 = \Theta(1)$. For all subsequent steps $t \geq 1$, the gradient ∇_{B_t} becomes non-zero. Under the same asymptotic scaling convention, multiplying by A_t^T does not introduce an additional width-dependent factor, so the effective gradient scale for B_l remains order-stable:

$$\|\nabla_{B_t}\|_F \asymp \left\|\nabla_{W_t} A_t^T\right\|_F = \Theta(1) \quad (\text{for } t \geq 1). \tag{20}$$

To achieve our goal of $\Delta B_l = \Theta(1)$, the learning rate η_{B_l} must be:

$$\Delta B_l = \eta_{B_l} \cdot \nabla_{B_l} \implies \Theta(1) = \eta_{B_l} \cdot \Theta(1) \implies \eta_{B_l} = \Theta(1). \tag{21}$$

The analysis therefore suggests, under this adopted scaling view rather than as a sharp matrix-norm bound, that the asymptotic scaling requires separating the learning rates:

$$\eta_{A_l} = \Theta(1/n) \quad \text{and} \quad \eta_{B_l} = \Theta(1). \tag{22}$$

This directly suggests that the corresponding learning rate ratio $\lambda = \eta_{B_l}/\eta_{A_l}$ should scale with the network width, $\lambda \propto n$. This principle aligns with broader findings in adaptive optimization, where per-parameter learning rates have proven highly effective [2, 29]. Our key insight is that this global width n should be replaced by a layer-specific effective width, which we measure using the stable rank of pre-trained weights.

B Justification for Stable Rank as a Complexity Metric

A central step in our framework is the replacement of the global architectural width n with a layer-specific effective width $n_{\text{eff}}(l)$. We argue that the Stable Rank of the pre-trained weight matrix W_l^* is a principled choice for this quantity [39]. We provide three justifications.

1. Connection to Effective Dimensionality. The singular value decomposition (SVD) of $W_l^* = U_l \Sigma_l V_l^T$ decomposes the linear transformation into a series of one-dimensional scalings along orthogonal bases. The squared singular values, σ_i^2, represent the energy captured by each singular vector pair. The Stable Rank, $\mathrm{srank}(W_l^*) = (\sum_i \sigma_i^2)/\sigma_{\max}^2$, can be interpreted as the number of dimensions a uniform-spectrum matrix would need to possess the same total energy as W_l^* while maintaining the same spectral norm. It therefore measures the effective number of degrees of freedom utilized by the matrix, providing a much finer-grained complexity metric than nominal rank.

2. Perturbation Sensitivity and Optimization Stability. Consider the LoRA update $\Delta W_l = B_l A_l$ as a perturbation. The appropriate update magnitude depends on the condition number and geometry of W_l^*:

- **Low Stable Rank (Anisotropic / Sharp):** A low stable rank implies that the matrix energy is concentrated in a few dominant directions ($\sigma_{\max} \gg \sigma_{others}$). In these directions, the optimization landscape is sharp (high curvature). While these directions are expressive, they act as stability bottlenecks: a large update projection along these dominant singular vectors can cause the layer's output to diverge drastically. Therefore, a smaller multiplier μ_l is generally required to satisfy local Lipschitz constraints and ensure stability.
- **High Stable Rank (Isotropic / Flat):** A high stable rank indicates a more uniform spectrum where energy is distributed across many dimensions. The transformation is isotropic, and the optimization landscape is relatively flat. In such a distributed system, adaptation signals are prone to dilution (as argued in Section 3.2). To achieve meaningful feature learning and escape the "lazy regime" in this high-dimensional subspace, a larger perturbation magnitude is required. This justifies a higher multiplier μ_l.

Thus, the stable rank serves as a dual indicator: it signals stability constraints (low rank) and feature learning requirements (high rank).

3. Adaptation vs. Preservation: Why PEFT Differs from Compression. A potential concern is that in model compression literature, high stable rank layers are often preserved with minimal changes to avoid accuracy loss [64]. However, this reasoning applies to the preservation objective. In contrast, PEFT operates under an adaptation objective: we explicitly aim to modify the model's behavior to fit a new downstream task [33, 43]. In this context:

- High stable rank indicates that the layer's representational capacity is distributed. To meaningfully adapt such distributed representations, the update must be large enough to perturb the entire active subspace; otherwise, the layer remains frozen.
- Our method prescribes larger adaptation intensity precisely to ensure that high-complexity layers can theoretically escape the lazy regime and learn task-specific features.

Furthermore, regarding the concern that "pure noise" also yields high stable rank: First, layers in well-trained models (e.g., Llama-2, RoBERTa) encode structured, non-random transformations learned from massive corpora [13, 60]. Second, even in pathological cases, our hybrid calibration mechanism (Section 3.4) acts as a corrective filter: a noisy layer would exhibit low task-specific gradient energy, resulting in a reduced multiplier regardless of its structural prior. The two-stage design ensures robustness.

C Justification for Gradient Energy as a Task-Relevance Proxy

Our hybrid calibration mechanism utilizes the squared Frobenius norm of the layer-wise gradient, which we term gradient energy, as a measure of task relevance. This choice is theoretically grounded in its connection to the Fisher Information Matrix (FIM) [39, 46].

For a model with parameters θ, the FIM, $\mathcal{F}$, is defined as the expectation of the outer product of the score function (gradient of the log-likelihood). Its entries are given by:

$$\mathcal{F}_{ij} = \mathbb{E}_{x \sim p(x)} \left[\frac{\partial \log p(y|x,\theta)}{\partial \theta_i} \frac{\partial \log p(y|x,\theta)}{\partial \theta_j} \right]. \tag{23}$$

The trace of the FIM, $\mathrm{Tr}(\mathcal{F}) = \sum_i \mathcal{F}_{ii}$, represents the total sensitivity of the model's predictive distribution to infinitesimal changes in parameters. It serves as a fundamental measure of the "amount of information" the data provides about the parameters.

For a specific layer l, the trace of its corresponding block, $\mathrm{Tr}(\mathcal{F}_l)$, quantifies the total importance of that layer's weights for the given task. In practical optimization, assuming the loss function $\mathcal{L}$ corresponds to the negative log-likelihood (e.g., Cross-Entropy), computing the exact FIM is intractable. However, we can approximate its trace using the *Empirical Fisher*. For notation, we write the proxy in terms of the virtual LoRA update $\Delta W_l = B_l A_l$. For a single data point (x, y), the squared Frobenius norm of the corresponding gradient is:

$$\left\| \frac{\partial \mathcal{L}(y, f(x;\theta))}{\partial \Delta W_l} \right\|_F^2 = \sum_{i,j} \left(\frac{\partial \mathcal{L}}{\partial (\Delta W_l)_{ij}} \right)^2. \tag{24}$$

Mathematically, this sum of squared gradients is equivalent to the trace of the rank-1 Empirical Fisher Information Matrix (EFIM) for that sample. Therefore, by accumulating this gradient energy $E_l = \sum_t \|\nabla_{\Delta W_l} \mathcal{L}_t\|_F^2$ over the calibration period, we obtain a practical proxy for the relative importance of each layer under the Fisher-information perspective. In implementation, this quantity is estimated in a memory-efficient manner from the available LoRA gradients rather than by explicitly materializing the dense gradient for ΔW_l. This provides a principled, data-driven signal for allocating adaptation intensity, justifying its use as the dynamic feedback component in our hybrid framework.

D Formal Properties of the Stable Rank Metric

We formally establish two key properties of the Stable Rank that justify its use as a robust complexity metric [12, 33].

Lemma D.1 (Basic Properties and Perturbation Stability of Stable Rank)**.** *For any matrix M, the stable rank, $\mathrm{srank}(M)$, satisfies:*

- ***(i) Bounds:*** $1 \leq \mathrm{srank}(M) \leq \mathrm{rank}(M)$.
- ***(ii) Robustness:*** *If a perturbation Δ satisfies $\|\Delta\|_2 \leq \epsilon \|M\|_2$ and $\|\Delta\|_F \leq \epsilon \|M\|_F$ for some $\epsilon < 1$, then the stable rank of the perturbed matrix is close to the original:* $\mathrm{srank}(M+\Delta) = \mathrm{srank}(M) \cdot (1 + O(\epsilon))$.

Proof. Let $r = \mathrm{rank}(M)$, and let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ be the non-zero singular values of M. By definition, $\|M\|_F^2 = \sum_{i=1}^r \sigma_i^2$ and $\|M\|_2 = \sigma_{\max}(M) = \sigma_1$.

(i) Bounds: For the upper bound, we use the fact that $\sigma_i \leq \sigma_1$ for all i:

$$\|M\|_F^2 = \sum_{i=1}^r \sigma_i^2 \leq \sum_{i=1}^r \sigma_1^2 = r \cdot \sigma_1^2 = \mathrm{rank}(M) \cdot \|M\|_2^2. \tag{25}$$

Dividing by $\|M\|_2^2$, we get $\mathrm{srank}(M) \leq \mathrm{rank}(M)$. For the lower bound, the sum of all squared singular values must be at least as large as the largest one:

$$\|M\|_F^2 = \sum_{i=1}^r \sigma_i^2 \geq \sigma_1^2 = \|M\|_2^2. \tag{26}$$

Dividing by $\|M\|_2^2$, we get $\mathrm{srank}(M) \geq 1$.

(ii) Robustness: Let $\tilde{M} = M + \Delta$. We are given $\|\Delta\|_2 \leq \epsilon \|M\|_2$ and $\|\Delta\|_F \leq \epsilon \|M\|_F$. We need to bound $\mathrm{srank}(\tilde{M}) = \frac{\|\tilde{M}\|_F^2}{\|\tilde{M}\|_2^2}$.

First, we bound the numerator $\|\tilde{M}\|_F^2$ using the triangle inequality:

$$\|\tilde{M}\|_F = \|M + \Delta\|_F \leq \|M\|_F + \|\Delta\|_F \leq \|M\|_F + \epsilon \|M\|_F = (1+\epsilon)\|M\|_F, \tag{27}$$

$$\|\tilde{M}\|_F = \|M + \Delta\|_F \geq \|M\|_F - \|\Delta\|_F \geq \|M\|_F - \epsilon \|M\|_F = (1-\epsilon)\|M\|_F. \tag{28}$$

Squaring these gives $\|\tilde{M}\|_F^2 = (1 \pm O(\epsilon))\|M\|_F^2$.

Second, we bound the denominator $\|\tilde{M}\|_2^2$ using Weyl's inequality for singular values, which states $|\sigma_i(\tilde{M}) - \sigma_i(M)| \leq \|\Delta\|_2$. For the largest singular value ($i = 1$):

$$\|\tilde{M}\|_2 = \sigma_{\max}(\tilde{M}) \leq \sigma_{\max}(M) + \|\Delta\|_2 \leq \|M\|_2 + \epsilon\|M\|_2 = (1+\epsilon)\|M\|_2, \tag{29}$$

$$\|\tilde{M}\|_2 = \sigma_{\max}(\tilde{M}) \geq \sigma_{\max}(M) - \|\Delta\|_2 \geq \|M\|_2 - \epsilon\|M\|_2 = (1-\epsilon)\|M\|_2. \tag{30}$$

Squaring these gives $\|\tilde{M}\|_2^2 = (1 \pm O(\epsilon))\|M\|_2^2$.

Finally, we combine these bounds to find the ratio:

$$\text{srank}(\tilde{M}) = \frac{\|\tilde{M}\|_F^2}{\|\tilde{M}\|_2^2} \leq \frac{(1+O(\epsilon))\|M\|_F^2}{(1-O(\epsilon))\|M\|_2^2} = \left(\frac{1+O(\epsilon)}{1-O(\epsilon)}\right)\text{srank}(M). \tag{31}$$

Since $\frac{1}{1-x} = 1 + x + O(x^2)$, the term $(1+O(\epsilon))(1-O(\epsilon))^{-1} = (1+O(\epsilon))(1+O(\epsilon)) = 1+O(\epsilon)$. A similar argument provides the lower bound:

$$\text{srank}(\tilde{M}) \geq \frac{(1-O(\epsilon))\|M\|_F^2}{(1+O(\epsilon))\|M\|_2^2} = (1-O(\epsilon))\text{srank}(M). \tag{32}$$

Combining the upper and lower bounds, we conclude that $\text{srank}(M + \Delta) = \text{srank}(M) \cdot (1 + O(\epsilon))$. □

E Mathematical Rationale for Structure-Aware Scaling

Here, we provide a theoretical rationale for replacing the global network width n with the layer-specific Stable Rank in the scaling laws of LoRA. The analysis in this section should be interpreted as a first-order modeling argument rather than an exact curvature law for arbitrary Transformer layers.

Lemma E.1 (Spectral Correspondence between Weights and Curvature)**.** *Consider the local quadratic approximation of the loss $\Delta\mathcal{L} \approx \frac{1}{2}Tr((\Delta W_l)^T \mathcal{H}_l \Delta W_l)$, where $\mathcal{H}_l$ denotes a local PSD curvature approximation for layer l (e.g., the Hessian in a locally convex neighborhood or a Gauss–Newton-type surrogate). The effective rank of the loss curvature, which measures the ratio of total sensitivity to peak sensitivity, is defined as:*

$$\text{EffectiveRank}(\mathcal{H}_l) = \frac{\text{Tr}(\mathcal{H}_l)}{\|\mathcal{H}_l\|_2} = \frac{\sum_i \lambda_i(\mathcal{H}_l)}{\lambda_{\max}(\mathcal{H}_l)}. \tag{33}$$

In converged models, the dominant singular directions of the weight matrix $W_l^\star$ encode the most critical features for signal propagation. Recent theoretical and empirical analyses on high-dimensional learning dynamics [44, 54] suggest a qualitative link between the spectral properties of pre-trained weights and the local Hessian geometry. However, for full Transformer layers with LayerNorm, residual connections, attention masking, and nonlinear interactions, we do not claim that this link yields exact closed-form identities. Instead, inspired by these prior analyses, we introduce the following first-order modeling assumptions to transfer the spectral-curvature correspondence into our setting and motivate a layer-wise structural prior:

1. *The total sensitivity (trace) of the Hessian scales with the total energy (squared Frobenius norm) of the weights:*

$$\text{Tr}(\mathcal{H}_l) = \alpha_l \|W_l^\star\|_F^2 = \alpha_l \sum_i \sigma_i(W_l^\star)^2, \tag{34}$$

   where $\alpha_l > 0$ is a layer-dependent proportionality constant.

2. *The peak sensitivity (max eigenvalue) of the Hessian scales with the peak energy (squared spectral norm) of the weights, consistent with prior closed-form analysis [54]:*

$$\|\mathcal{H}_l\|_2 = \beta_l \|W_l^\star\|_2^2 = \beta_l \sigma_{\max}(W_l^\star)^2, \tag{35}$$

   where $\beta_l > 0$ is a second layer-dependent proportionality constant.

Under these assumptions, the effective rank of the Hessian becomes:

$$\mathrm{EffectiveRank}(\mathcal{H}_l) = \frac{\mathrm{Tr}(\mathcal{H}_l)}{\|\mathcal{H}_l\|_2} \approx \frac{\alpha_l}{\beta_l} \cdot \frac{\|W_l^\star\|_F^2}{\|W_l^\star\|_2^2} = \gamma_l \,\mathrm{srank}(W_l^\star), \tag{36}$$

where $\gamma_l \triangleq \alpha_l/\beta_l$. Eq. (36) *is therefore a conditional approximation rather than an exact identity: stable rank enters the argument as a first-order structural signal modulated by an unknown layer-dependent factor. Our intended scope is comparable Transformer blocks within the same backbone, where we treat γ_l as a same-order working approximation rather than as an exactly shared constant. When the ratios γ_l vary mildly across layers, stable rank preserves the dominant ordering signal; when they vary substantially, a purely structural allocation can over- or under-estimate the ideal multiplier for individual layers. This failure mode is precisely why SA-LoRA does not rely on the static prior alone: the hybrid gradient calibration stage is designed to correct residual layer-wise mismatch online.*

Based on Lemma E.1, we can derive, within this modeling framework, the relationship between the stable rank and the adaptation intensity suggested by the stability constraints of gradient descent.

Corollary E.2 (Stable Rank-Guided Stability Bound)**.** *For gradient-based optimization on a β-smooth loss landscape, the learning rate (or adaptation intensity) η_l for layer l is theoretically bounded by the inverse of the spectral norm of the Hessian: $\eta_l \leq \frac{2}{\lambda_{\max}(\mathcal{H}_l)}$.*

By substituting the relationship from Lemma E.1, specifically $\lambda_{\max}(\mathcal{H}_l) = \frac{\mathrm{Tr}(\mathcal{H}_l)}{\mathrm{EffectiveRank}(\mathcal{H}_l)}$, into this stability bound, we obtain:

$$\eta_l \leq \frac{2 \cdot \mathrm{EffectiveRank}(\mathcal{H}_l)}{\mathrm{Tr}(\mathcal{H}_l)} \approx \frac{2\gamma_l \,\mathrm{srank}(W_l^\star)}{\mathrm{Tr}(\mathcal{H}_l)}. \tag{37}$$

Crucially, while Eq. (34) indicates that the raw trace $\mathrm{Tr}(\mathcal{H}_l)$ scales with weight energy $\|W_l^\star\|_F^2$, modern adaptive optimizers (e.g., Adam) partially normalize gradient magnitudes through moment-based preconditioning. This reduces the dominance of the global sensitivity term $\mathrm{Tr}(\mathcal{H}_l)$ in the effective step size. In this normalized optimization regime, the variation in optimal step size is no longer dominated by the gross magnitude ($\mathrm{Tr}(\mathcal{H}_l)$), but rather by the structural anisotropy of the layer together with the residual factor γ_l. Consequently, the maximum stable adaptation intensity is approximately ordered by $\gamma_l \mathrm{srank}(W_l^\star)$:

$$\eta_l^{\max} \propto \gamma_l \,\mathrm{srank}(W_l^\star). \tag{38}$$

Because γ_l is unknown in practice, SA-LoRA does not rely on stable rank as an exact layer-wise optimum. Instead, it uses stable rank as an initialization of the allocation prior, after which budget-conserving normalization controls the global scale and hybrid gradient calibration corrects layer-wise mismatches induced by variation in γ_l.

This derivation provides a theory-motivated justification for our structure-aware scaling under the assumptions above. Our claim is therefore not that stable rank exactly determines curvature or the optimal multiplier for arbitrary Transformer layers, but that it offers a practically useful first-order structural prior for layer-wise allocation. We further emphasize that budget-conserving normalization does not certify the correctness of the layer-wise ranking; it only guarantees that the allocation remains within a safe global budget. This interpretation is consistent with the empirical gains of the structural-only variant in Table 5, while the budget-conserving normalization and hybrid gradient calibration further reduce sensitivity to possible prior misspecification and compensate for unknown variation in γ_l. By linking the stable rank to the stability bound of the optimization in this approximate sense, we can interpret the dynamics as follows:

- **Low Stable Rank (High Sharpness $\rightarrow$ Small Updates):** A low stable rank ($\mathrm{srank} \to 1$) implies that the Hessian energy is concentrated in a few dominant principal components (highly anisotropic). This results in a large $\lambda_{\max}(\mathcal{H}_l)$, creating a "sharp" loss landscape where gradient descent operates at the edge of stability [6]. To ensure stability and prevent divergence along these sensitive directions, the adaptation intensity η_l must be strictly capped.

- **High Stable Rank (Low Sharpness → Large Updates):** A high stable rank indicates a distributed spectrum with no single dominant direction of extreme curvature (small $\lambda_{\max}(\mathcal{H}_l)$). This implies an isotropic and relatively "flat" landscape. In such regimes, standard step sizes are overly conservative. To achieve meaningful feature learning and escape the "lazy regime" across the many active dimensions, the adaptation intensity can be scaled up proportionally to the stable rank without violating stability constraints.

Combining the formal stability bound (Corollary E.2) with the feature learning requirements leads to our central proposition regarding the layer-wise adaptation magnitude.

Proposition E.3 (Spectrally-Guided Adaptation Intensity)**.** *Let μ_l be the adaptation intensity multiplier for layer l, governing the effective step size of the optimization trajectory. The first-order structural prior suggested by this analysis is to initialize the layer-wise allocation according to:*

$$\mu_l \propto \operatorname{srank}(W_l^\star). \tag{39}$$

This proposition should be interpreted as a principled initialization rule rather than an exact optimum, with the subsequent gradient-based calibration compensating for unknown layer-wise deviations such as γ_l.

F Formal Justification for Gradient Energy Proxy

This section justifies our use of accumulated gradient energy as a measure of a layer's importance to the downstream task.

Let θ_l be the set of LoRA parameters for layer l. We are interested in the trace of the Fisher Information Matrix (FIM) [52] for these parameters, $\operatorname{tr}(\mathcal{I}(\theta_l))$, as a measure of task importance. We posit that this intractable quantity is proportional to the expected squared norm of the stochastic gradient.

Proposition F.1 (Gradient Energy and Fisher Information Trace)**.** *Under standard assumptions, the expected squared L_2 norm of the stochastic gradient for a mini-batch is equal to the trace of the FIM for that layer's parameters:*

$$\mathbb{E}\left[\|\nabla_{\theta_l}\mathcal{L}\|_2^2\right] = \operatorname{tr}(\mathcal{I}(\theta_l)). \tag{40}$$

Proof. The Fisher identity states $\mathcal{I}(\theta_l) = \mathbb{E}[(\nabla_{\theta_l}\mathcal{L})(\nabla_{\theta_l}\mathcal{L})^\top]$. The trace of the FIM is therefore the expected squared L_2 norm of the gradient, proving the proposition:

$$\operatorname{tr}(\mathcal{I}(\theta_l)) = \operatorname{tr}(\mathbb{E}[(\nabla_{\theta_l}\mathcal{L})(\nabla_{\theta_l}\mathcal{L})^\top]) = \mathbb{E}[\operatorname{tr}((\nabla_{\theta_l}\mathcal{L})(\nabla_{\theta_l}\mathcal{L})^\top)] = \mathbb{E}[\|\nabla_{\theta_l}\mathcal{L}\|_2^2]. \tag{41}$$

However, in our method, we do not compute $\nabla_{\theta_l}\mathcal{L}$ directly. Instead, for notation we write $G_W = \nabla_{\Delta W_l}\mathcal{L}$, the gradient with respect to the virtual low-rank update $\Delta W_l = B_l A_l$. The gradient energy we accumulate is $E_l = \sum_t \|G_W^{(t)}\|_F^2$, which is an empirical estimate of $\mathbb{E}[\|G_W\|_F^2]$. In implementation, however, $G_W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ is not explicitly materialized; the corresponding layer-wise energy is estimated in a memory-efficient manner from the available LoRA gradients.

This quantity serves as a practical, computationally efficient proxy for the true trace $\operatorname{tr}(\mathcal{I}(\theta_l))$. The justification for this proxy is that the two gradient norms are fundamentally coupled via the chain rule (as seen in Eq. (2)):

$$G_A = \nabla_{A_l}\mathcal{L} = B_l^T G_W, \tag{42}$$

$$G_B = \nabla_{B_l}\mathcal{L} = G_W A_l^T. \tag{43}$$

Crucially, even at initialization where $B_l = 0$ (implying $G_A = 0$), the term G_B remains non-zero due to the random initialization of A_l, ensuring that meaningful gradient energy accumulation commences immediately at the first step.

The true quantity of interest is $\operatorname{tr}(\mathcal{I}(\theta_l)) = \mathbb{E}[\|G_A\|_F^2 + \|G_B\|_F^2]$. Since the magnitudes of A_l and B_l are relatively stable during fine-tuning (remaining $\Theta(1)$ as per the scaling laws in Appendix A), the norm of the input gradient $\|G_W\|_F^2$ is strongly correlated with the norm of the output gradients $\|G_A\|_F^2 + \|G_B\|_F^2$. Therefore, using the easily computable $\mathbb{E}[\|G_W\|_F^2]$ as a proxy for $\operatorname{tr}(\mathcal{I}(\theta_l))$ is a principled approach to estimate the relative task sensitivity of the layer. □

Thus, the accumulated early-stage gradient energy E_l serves as a sensible proxy for the task-relevance of layer l. Our implementation of the hybrid calibration mechanism (controlled by `sa_grad_calibrate_steps`, `sa_grad_power`, and `sa_grad_blend`) directly operationalizes this insight by mapping these energy scores into learning rate multipliers, thereby steering the learning budget towards the most task-sensitive layers [4].

G Stability Analysis of the Online Algorithm

For the purpose of theoretical analysis, we model the dynamic update process described in Algorithm 1 using a bounded update framework. While the practical implementation applies a power transformation followed by budget-conserving normalization, for stability analysis we study a bounded-update approximation of the practical dynamics. Specifically, since the power transformation bounds the preliminary multipliers within $[\mu_{\min}, \mu_{\max}]$ and budget conservation enforces $\frac{1}{L}\sum \mu_l = 1$, the final multipliers μ_l remain within the bounded range $[\mu_{\min}/\mu_{\max}, \mu_{\max}/\mu_{\min}]$ at every step. The analysis in this section therefore addresses safety and boundedness of the update process; it does not establish that budget conservation alone recovers the correct layer-wise ranking. Based on this modeling assumption, we provide two lemmas to explain the observed stability.

Lemma G.1 (Damping Effect of EMA)**.** *Let the multiplier update rule $\mu^{(t)} = \Psi(E^{(t)})$ be L_Ψ-Lipschitz continuous, and let the energy $E^{(t)}$ be updated via the Exponential Moving Average (EMA) rule: $E^{(t)} = \beta_{\text{ema}} E^{(t-1)} + (1-\beta_{\text{ema}})G^{(t)}$. If the instantaneous gradient energy is bounded, i.e., $\|G^{(t)}\|_\infty \le G_{\max}$, then the change in multipliers between consecutive steps is bounded by:*

$$\|\mu^{(t)} - \mu^{(t-1)}\|_\infty \le L_\Psi (1-\beta_{\text{ema}}) G_{\max}. \tag{44}$$

Proof. We begin by exploiting the L_Ψ-Lipschitz property of the mapping function Ψ:

$$\|\mu^{(t)} - \mu^{(t-1)}\|_\infty = \|\Psi(E^{(t)}) - \Psi(E^{(t-1)})\|_\infty \le L_\Psi \|E^{(t)} - E^{(t-1)}\|_\infty. \tag{45}$$

Next, we analyze the perturbation in the accumulated energy $E^{(t)}$. By the EMA definition:

$$E^{(t)} - E^{(t-1)} = \left(\beta_{\text{ema}} E^{(t-1)} + (1-\beta_{\text{ema}})G^{(t)}\right) - E^{(t-1)} \tag{46}$$

$$= (1-\beta_{\text{ema}})(G^{(t)} - E^{(t-1)}). \tag{47}$$

Taking the infinity norm yields:

$$\|E^{(t)} - E^{(t-1)}\|_\infty = (1-\beta_{\text{ema}})\|G^{(t)} - E^{(t-1)}\|_\infty. \tag{48}$$

Since $E^{(t-1)}$ is a convex combination of past gradient energies (all within $[0, G_{\max}]$), its value implies $E^{(t-1)} \in [0, G_{\max}]$. Consequently, the maximum deviation between the instantaneous gradient $G^{(t)}$ and the smoothed history $E^{(t-1)}$ is bounded by the maximum scale $G_{\max}$:

$$\|G^{(t)} - E^{(t-1)}\|_\infty \le \max(G_{\max} - 0, 0 - G_{\max}) = G_{\max}. \tag{49}$$

Substituting this back into the Lipschitz inequality proves the lemma:

$$\|\mu^{(t)} - \mu^{(t-1)}\|_\infty \le L_\Psi (1-\beta_{\text{ema}}) G_{\max}. \tag{50}$$

This result formally demonstrates that a larger decay factor β_{ema} (closer to 1) linearly damps the maximum per-step oscillation of the multipliers, ensuring optimization stability. □

Lemma G.2 (Safety under Budget Conservation)**.** *Consider a set of preliminary multipliers m_l derived from the structural and gradient scores. Let the final multipliers $\mu_l^{(t)}$ be obtained by enforcing the budget constraint $\frac{1}{L}\sum \mu_l^{(t)} = 1$ via normalization. If the preliminary multipliers are bounded such that $m_l \in [\mu_{\min}, \mu_{\max}]$ (e.g., via the power transformation), then the global scaling factor remains stable, preventing gradient explosion.*

Proof. The budget-conserving normalization is defined as $\mu_l^{(t)} = m_l / \bar{m}$, where $\bar{m} = \frac{1}{L}\sum m_k$. Since each preliminary multiplier is bounded $\mu_{\min} \le m_l \le \mu_{\max}$, the mean is also bounded:

$$\mu_{\min} \le \bar{m} \le \mu_{\max}. \tag{51}$$

Consequently, the normalization factor $1/\bar{m}$ is bounded within $[1/\mu_{\max}, 1/\mu_{\min}]$. This implies that the budget conservation step does not arbitrarily inflate the multipliers. Specifically, the final multipliers satisfy:

$$\frac{\mu_{\min}}{\mu_{\max}} \leq \mu_l^{(t)} \leq \frac{\mu_{\max}}{\mu_{\min}}. \tag{52}$$

Given typical hyperparameters (e.g., $\mu_{\min} = 0.2, \mu_{\max} = 2.0$), the ratio is bounded. This guarantees that enforcing the global budget $\frac{1}{L}\sum \mu_l^{(t)} = 1$ keeps individual layer multipliers within a safe, bounded operating regime, effectively preventing the "runaway" adaptation that can occur in unconstrained dynamic methods. Importantly, this lemma is a safety statement: it ensures feasibility and boundedness, but does not imply that budget conservation alone produces the correct layer-wise ordering or spacing of multipliers. □

Together, Lemma G.1 and Lemma G.2 provide a theoretical basis for the stability observed in our experiments: the former ensures smoothness over time, while the latter ensures boundedness under a fixed global budget. The layer-wise correctness of the allocation is instead only approximate and is improved in practice by the hybrid calibration mechanism.

H Experimental Setup Details

H.1 Dataset Details

We do not use a single target-module configuration across all benchmarks. For RoBERTa-based NLU and GPT-2-based NLG experiments, we apply LoRA to the Query and Value projections. For instruction-tuning on Llama/Llama2, we follow a broader target-module setting covering K/Q/V/O together with the MLP up, down, and gate projections. For the supplementary multimodal VL-BART experiment, we adapt the Query and Key projections. We conduct experiments across four evaluation scenarios. The specific tasks and their evaluation metrics are detailed in Table 7. Beyond the evaluated benchmarks, broader multimodal applications encompass compositional visual understanding [23, 63, 70] and neural visual decoding [24, 25].

Table 7: Dataset and evaluation metric details.

BENCHMARK	TASK	METRIC
GLUE	MNLI	MATCHED/MISMATCHED ACC.
	QQP	ACCURACY
	SST-2	ACCURACY
	QNLI	ACCURACY
	CoLA	MATTHEWS CORR.
	RTE	ACCURACY
	MRPC	ACCURACY
	STS-B	PEARSON/SPEARMAN CORR.
E2E NLG	DATA-TO-TEXT	BLEU NIST METEOR ROUGE-L CIDER
MT-BENCH	INSTRUCTION FOLLOWING	GPT-4 SCORE

H.2 Hyperparameter Settings

To ensure reproducibility, we provide the detailed hyperparameters used for our main results. All experiments are conducted on NVIDIA A800 80GB GPUs. We follow the standard hyperparameter configurations from prior work [16, 19] and only tune task-specific parameters. Table 8 shows settings for NLU tasks, Table 9 for NLG tasks, Table 10 for instruction-following tasks, and Table 11 for the supplementary multimodal evaluation.

Table 8: Hyperparameter configurations for GLUE benchmark.

Model	Hyperparameter	MNLI	QQP	SST-2	QNLI	CoLA	RTE	MRPC	STS-B
Base	Optimizer	AdamW							
	adam_beta1	0.9							
	adam_beta2	0.99							
	Weight Decay	0.01							
	LoRA Rank (r)	8							
	LoRA Alpha (α)	8							
	Epochs	10	10	40	30	40	40	30	40
	Learning Rate	2E-4	2E-4	2E-4	2E-4	2E-4	2E-4	2E-4	2E-4
	Batch Size	64							
Large	Optimizer	AdamW							
	adam_beta1	0.9							
	adam_beta2	0.99							
	Weight Decay	0.01							
	LoRA Rank (r)	8							
	LoRA Alpha (α)	8							
	Epochs	10	10	20	20	30	30	20	30
	Learning Rate	3E-4	4E-4	2E-4	2E-4	2E-4	3E-4	2E-4	2E-4
	Batch Size	32							
SA-LoRA Specific									
	$\mu_{\min}$, $\mu_{\max}$	0.4, 3.2							
	Power p	2.5							
	sa_grad_blend	0.2							

Table 9: Hyperparameter configurations for the E2E NLG benchmark (GPT-2 Medium/Large).

Hyperparameter	Medium	Large
Optimizer	AdamW	
warmup_ratio	0.03	
Weight Decay	0.01	
LoRA Rank (r)	8	8
LoRA Alpha (α)	16	16
Learning Rate	2E-4	1E-4
Batch Size	16	8
Epochs	8	8
SA-LoRA Specific		
$\mu_{\min}$, $\mu_{\max}$	0.4, 3.2	
Power p	2.5	
sa_grad_blend	0.2	

Table 10: Hyperparameter configurations for instruction-tuning (Llama/Llama2 7B/13B).

HYPERPARAMETER	7B	13B
OPTIMIZER	ADAMW	
WARMUP RATIO	0.03	
WEIGHT DECAY	0.01	
DROPOUT	0.05	
LORA RANK (r)	64	
LORA ALPHA (α)	64	
LEARNING RATE	2E-4	1E-4
BATCH SIZE	16	8
EPOCHS	3	3
SA-LoRA Specific		
$\mu_{\min}$, $\mu_{\max}$	0.4, 3.2	
POWER p	2.5	
SA_GRAD_BLEND	0.2	

Table 11: Hyperparameter configurations for the supplementary multimodal evaluation (VL-BART).

HYPERPARAMETER	VALUE
OPTIMIZER	ADAMW
LR SCHEDULER	LINEAR
LORA RANK (r)	128
LORA ALPHA (α)	128
LEARNING RATE	1E-3
SA-LoRA Specific	
$\mu_{\min}$, $\mu_{\max}$	0.4, 3.2
POWER p	2.5
SA_GRAD_BLEND	0.2

I Training Cost Analysis

Compared to standard LoRA and LoRA+, SA-LoRA incurs only a minimal additional overhead (around 1%) due to the structure-aware coefficient updates via spectral-gradient fusion. The stable-rank preprocessing is a one-time step performed before fine-tuning. We compare the training cost in Table 12, 13, and 14.

It is worth noting that the memory consumption for GPT-2 models (Table 13) appears higher than that of Llama models (Table 14) in some cases. This is due to the different standard configurations used for each model family to ensure fair comparison with prior baselines. Specifically, GPT-2 experiments follow the standard FP32 precision without gradient checkpointing setting [19], whereas Llama experiments utilize the memory-efficient BF16 precision with gradient checkpointing.

Table 12: Training cost comparison on GLUE/SST-2 (NVIDIA A800 80GB GPU).

Model	Method	Memory (GB)	Time/Epoch (min)	Time Overhead
RoBERTa-base	LoRA	26.6G	8.90	-
	LoRA+	26.6G	8.97	+0.8%
	SA-LoRA	26.7G	8.99	+1.0%
RoBERTa-large	LoRA	35.0G	22.83	-
	LoRA+	35.0G	22.97	+0.6%
	SA-LoRA	35.1G	23.03	+0.9%

Table 13: Training cost comparison on E2E NLG (NVIDIA A800 80GB GPU).

Model	Method	Memory (GB)	Time/Epoch (min)	Time Overhead
GPT-2 Medium	LoRA	38.3G	31.02	-
	LoRA+	38.3G	31.12	+0.3%
	SA-LoRA	38.4G	31.25	+0.7%
GPT-2 Large	LoRA	35.5G	59.68	-
	LoRA+	35.5G	59.77	+0.2%
	SA-LoRA	35.7G	59.93	+0.4%

Table 14: Training cost comparison on Alpaca instruction-tuning (NVIDIA A800 80GB GPU).

Model	Method	Memory (GB)	Time/Epoch (h)	Time Overhead
Llama-7B	LoRA	33.4G	3.42	-
	LoRA+	33.4G	3.43	+0.3%
	SA-LoRA	33.5G	3.44	+0.6%
Llama-13B	LoRA	38.3G	5.03	-
	LoRA+	38.3G	5.04	+0.2%
	SA-LoRA	38.4G	5.06	+0.6%
Llama2-7B	LoRA	33.4G	3.42	-
	LoRA+	33.4G	3.43	+0.3%
	SA-LoRA	33.5G	3.44	+0.6%
Llama2-13B	LoRA	38.3G	5.03	-
	LoRA+	38.3G	5.04	+0.2%
	SA-LoRA	38.4G	5.06	+0.6%

J Supplementary Convergence Analysis

Figure 3 illustrates the training dynamics on the Llama-7B instruction-tuning setting. SA-LoRA not only achieves superior final performance but also converges faster than DoRA and LoRA. This empirical ordering is consistent with the optimization rationale in Appendix E: uniform LoRA is bottlenecked by the most sensitive layers and therefore uses an overly conservative allocation for flatter layers, while SA-LoRA redistributes the same global adaptation budget according to layer geometry, allowing larger useful updates where they are safe and smaller ones where sharpness demands caution. We therefore interpret Appendix E as explaining why SA-LoRA can achieve faster practical loss reduction, rather than as a complete formal convergence proof.

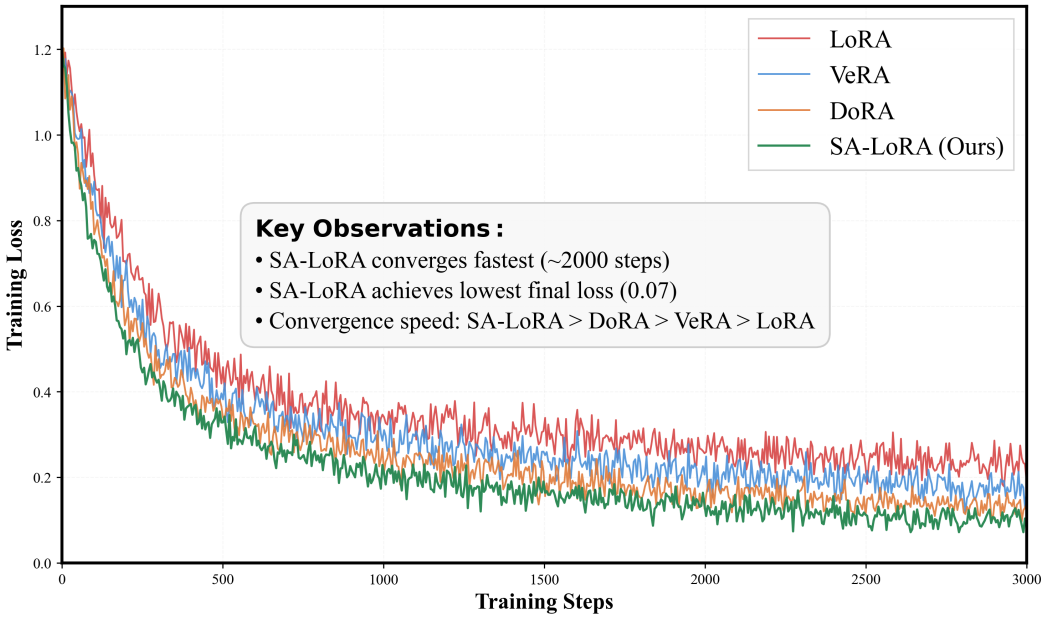

Figure 3: Training dynamics comparison on the Llama-7B setting. SA-LoRA converges faster and achieves better final performance than the baseline methods.

K Detailed Ablation Results

We provide the per-task breakdown of our component analysis ablation studies on RoBERTa-base. Table 15 details the performance evolution as we progressively incorporate the core mechanisms: Structure-Awareness, Budget Conservation, and Gradient Calibration. Consistent with the main text, the results confirm that the structural prior acts as the primary performance driver, while budget conservation and gradient calibration provide further stability and refinement. Table 16 reports the mapping power ablation, and Table 17 reports the hybrid calibration ablation.

Table 15: Detailed per-task results for the component analysis on RoBERTa-base. **Struct. Only** corresponds to using only the stable rank multiplier; **Static** includes budget conservation; **Full** adds hybrid gradient calibration.

METHOD	SST-2	MRPC	STS-B	AVG.
LORA (BASELINE)	95.1	89.7	91.5	92.1
SA-LORA (STRUCT. ONLY)	95.5	91.2	92.0	92.9
SA-LORA (STATIC)	95.6	91.6	92.1	93.1
SA-LORA (FULL)	95.7	92.0	92.1	93.3

Table 16: Detailed results for the mapping power ablation on RoBERTa-base.

MAPPING POWER p	SST-2	MRPC	STS-B	AVG.
$p = 0.5$ (CONCAVE)	95.3	90.8	91.4	92.5
$p = 1.0$ (LINEAR)	95.5	91.2	92.0	92.9
$p = 2.5$ (OURS)	95.7	92.0	92.1	93.3

Table 17: Detailed results for the hybrid calibration ablation on RoBERTa-base.

CALIBRATION STRATEGY	SST-2	MRPC	STS-B	AVG.
STRUCTURAL ONLY ($\beta = 0$)	95.5	91.9	91.8	93.1
GRADIENT ONLY ($\beta = 1$)	94.8	90.8	91.2	92.3
HYBRID ($\beta = 0.2$)	95.7	92.0	92.1	93.3

K.1 Inverted Allocation Ablation

To directly test whether the direction of our stable-rank-based allocation is meaningful, we construct an inverted allocation baseline on RoBERTa-base over SST-2, MRPC, and STS-B. In this variant, the score mapping is reversed so that lower-stable-rank layers receive larger multipliers and higher-stable-rank layers receive smaller ones, while keeping the same global budget and training protocol as SA-LoRA.

The results in Table 18 show that this reversed strategy performs substantially worse than the standard structure-aware allocation, and even below uniform LoRA. This provides direct supporting evidence that the direction of the stable-rank prior is not arbitrary: under the same budget, assigning larger multipliers to lower-stable-rank layers is detrimental, whereas allocating larger multipliers to higher-stable-rank layers is empirically more effective.

K.2 Alternative Allocation Signals

To test how much of the gain is specific to stable rank rather than to arbitrary layer-wise redistribution, we compare several intuitive allocation signals on RoBERTa-base over SST-2, MRPC, and STS-B under the same global budget and training protocol. In addition to the uniform baseline, we consider two scale-dependent alternatives based on the pre-trained weights: the Frobenius norm and the spectral norm. Stable rank corresponds to the ratio $\|W_l^\star\|_F^2 / \|W_l^\star\|_2^2$, which removes absolute scale and emphasizes structural anisotropy.

The results in Table 19 show that simple norm-based redistribution provides only marginal improvements over uniform LoRA, whereas stable rank yields a substantially larger gain. This comparison is

Table 18: Inverted-allocation ablation on RoBERTa-base over SST-2, MRPC, and STS-B.

Allocation Strategy	SST-2	MRPC	STS-B	Avg.
LoRA	95.1	89.7	91.5	92.1
Inverted SA-LoRA	93.8	88.2	91.0	91.0
SA-LoRA	95.7	92.0	92.1	**93.3**

not intended to be exhaustive, but it directly tests two natural scale-dependent alternatives and shows that the improvement is not obtained by arbitrary redistribution alone. Instead, the results support our interpretation that stable rank is a more effective structural prior because it captures geometric shape information beyond raw weight magnitude.

Table 19: Comparison with alternative allocation signals on RoBERTa-base over SST-2, MRPC, and STS-B.

Allocation signal	Formula	Avg. score
Standard	$\mu_l = 1$	92.1
Frobenius norm	$\mu_l \propto \|W_l^\star\|_F$	92.4
Spectral norm	$\mu_l \propto \|W_l^\star\|_2$	92.2
Stable rank (SA-LoRA)	$\mu_l \propto \|W_l^\star\|_F^2 / \|W_l^\star\|_2^2$	**93.3**

K.3 Stronger Gradient-Only Baseline

To test whether the gap between the gradient-only and hybrid variants is merely due to an under-tuned pure-gradient baseline, we construct a stronger gradient-only comparison on RoBERTa-base over SST-2, MRPC, and STS-B (3 seeds). In addition to the naive gradient-only variant reported in the main text, this strengthened version applies EMA smoothing to the layer-wise gradient statistics and budget-conserving normalization under the same multiplier range as SA-LoRA. This produces a more competitive pure-gradient allocation rule while keeping the rank budget fixed.

The results in Table 20 show that the stronger pure-gradient baseline substantially improves over the naive gradient-only extreme case (92.8 vs. 92.3), confirming that smoothing and normalization matter. However, it still remains below the structural-only variant (93.1) and the full hybrid model (93.3). This supports the interpretation that the stable-rank prior contributes genuine information beyond simply denoising a noisy gradient signal: it provides a task-agnostic anchor, while gradient feedback acts as a task-specific refinement.

Table 20: Comparison with a stronger pure-gradient baseline on RoBERTa-base over SST-2, MRPC, and STS-B (3 seeds).

Allocation Strategy	Configuration	Avg. Score
Uniform (Standard LoRA)	–	92.1
Naive Gradient-Only	No smoothing / normalization	92.3
Well-Tuned Gradient-Only	**EMA + normalization**	**92.8**
Structural Prior Only	$\beta = 0$	93.1
Hybrid (SA-LoRA)	**Prior + calibrated gradient**	**93.3**

K.4 Module Scope Ablation

To evaluate whether SA-LoRA depends on the standard Q/V adapter placement, we conduct a module-scope ablation on RoBERTa-base over SST-2, MRPC, and STS-B using the same training budget and protocol as the main experiments. We compare four target-module sets: (i) Q/V (our default setting), (ii) Q/K/V/O, (iii) Q/V + FFN, and (iv) Q/K/V/O + FFN. Since SA-LoRA is defined for arbitrary weight matrices, expanding the module set changes the candidate pool over which the fixed adaptation budget is redistributed; accordingly, the relative multipliers assigned to Q and V are allowed to shift when K, O, or FFN modules are introduced.

The results in Table 21 support three observations. First, extending attention adaptation from Q/V to Q/K/V/O preserves essentially the same average performance, indicating that SA-LoRA remains

stable under a broader attention-module scope. Second, enabling FFN on top of Q/V alone does not consistently improve performance, suggesting that naively increasing module coverage is not always beneficial under a fixed budget. Third, enabling both broader attention modules and FFN yields the best average result, which is consistent with SA-LoRA's intended behavior: structure-aware redistribution can exploit complementary modules when they provide useful adaptation capacity.

Table 21: Module-scope ablation on RoBERTa-base over SST-2, MRPC, and STS-B (3 seeds). Results are averaged across the three tasks.

TARGET MODULES	AVG. SCORE	STD.
Q, V	93.32	0.21
Q, K, V, O	93.34	0.30
Q, V + FFN	93.16	0.24
Q, K, V, O + FFN	**93.48**	0.28

L Dataset-Level Significance Tests

To improve the transparency of our empirical claims, we additionally report a seed-level significance check for the primary comparison between SA-LoRA and standard LoRA under the same backbone and parameter budget. Since all main experiments use three random seeds, we visualize the dataset-level Cohen's d effect sizes together with the corresponding two-sided Welch's t-test p-values. We emphasize that these results should be interpreted as small-sample inferential evidence rather than as definitive large-sample statistical conclusions.

Figure 4 visualizes the dataset-level results. The pattern is directionally consistent with the mean-performance trends in the main paper: SA-LoRA shows positive effect sizes on most datasets and metrics, with especially strong gains on several RoBERTa-base tasks, the main E2E generation metrics, and the 7B-scale MT-Bench settings. At the same time, some datasets remain statistically weak under the current small-sample setup, and one task (RTE on RoBERTa-base) shows an effect in the opposite direction. We therefore include these tests as supplementary transparency-oriented evidence rather than as a universal statistical guarantee.

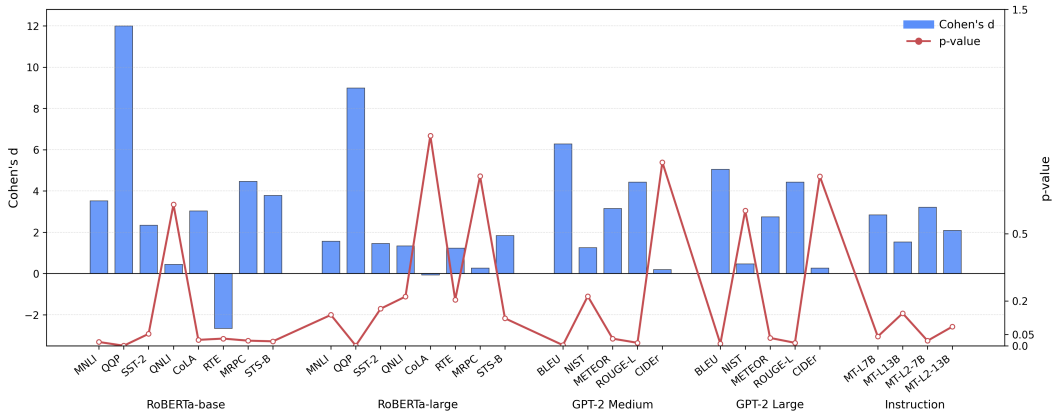

Figure 4: Dataset-level significance checks for SA-LoRA versus standard LoRA. Bars show Cohen's d effect sizes and the overlaid line shows the corresponding two-sided Welch's t-test p-values across evaluation settings.

M Impact of LoRA Rank (r)

To provide comprehensive evidence that SA-LoRA provides consistent gains regardless of the parameter budget, we compare standard LoRA and SA-LoRA across multiple LoRA ranks $r \in \{4, 8, 16, 32, 64\}$ on all eight GLUE tasks with RoBERTa-base. As shown in Figure 5, SA-LoRA consistently outperforms the baseline across all rank configurations and all datasets. The performance gap is more pronounced at lower ranks ($r = 4, 8$), particularly on challenging tasks like CoLA (+4.7% at $r = 4$), suggesting that structure-aware adaptation is especially beneficial when the parameter budget is strictly constrained.

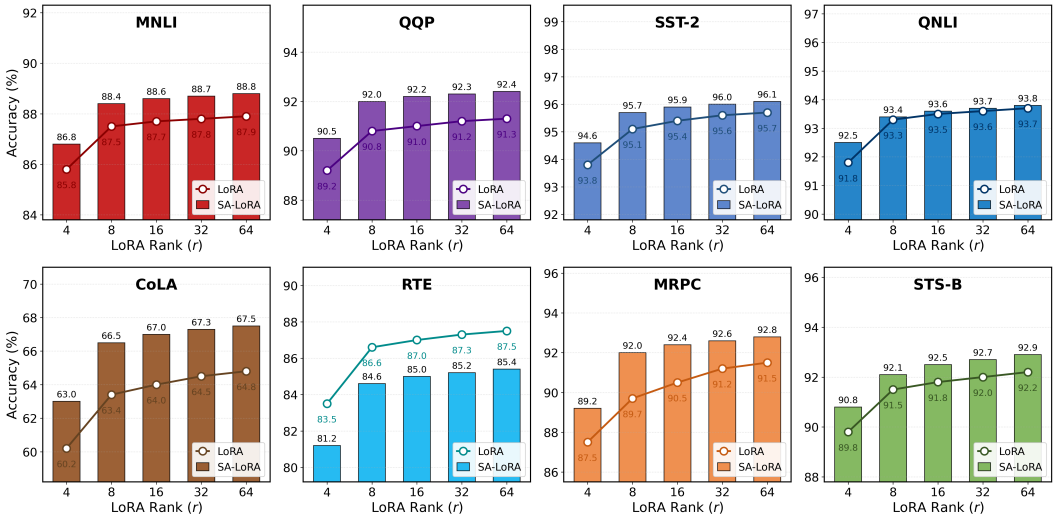

Figure 5: Complete performance comparison across different LoRA ranks (r) on all eight GLUE tasks (RoBERTa-base). SA-LoRA consistently outperforms standard LoRA (lines) across all datasets and rank configurations. The performance advantage is particularly pronounced at lower ranks, demonstrating the efficacy of structure-aware adaptation under constrained parameter budgets.

N Extended Hyperparameter Sensitivity Analysis

Figure 2 in the main paper showed the sensitivity to the LoRA rank r. Here, we provide a comprehensive analysis of SA-LoRA's sensitivity to its other key hyperparameters: the multiplier range $[\mu_{\min}, \mu_{\max}]$. We conduct experiments on RoBERTa-base by varying these parameters systematically on three GLUE tasks: SST-2, MRPC, and STS-B.

Sensitivity to Multiplier Range. Figure 6 shows the performance heatmaps when varying $\mu_{\min} \in \{0.2, 0.4, 0.6, 0.8\}$ and $\mu_{\max} \in \{1.5, 2.0, 2.5, 3.0, 3.5\}$. Across all three tasks, SA-LoRA exhibits consistent and robust behavior: performance improves as $\mu_{\max}$ increases, with the high-performance region spanning most configurations where $\mu_{\max} \geq 2.5$. Specifically, accuracy ranges from 93.2% to 95.7% on SST-2, 89.0% to 92.0% on MRPC, and 89.1% to 92.1% on STS-B. Our default setting ($\mu_{\min} = 0.4$, $\mu_{\max} = 3.2$, marked by $\star$) achieves strong or near-optimal performance on all tasks. These consistent patterns across tasks demonstrate that SA-LoRA's hyperparameter sensitivity is task-agnostic, and practitioners can confidently use a single configuration across diverse downstream tasks.

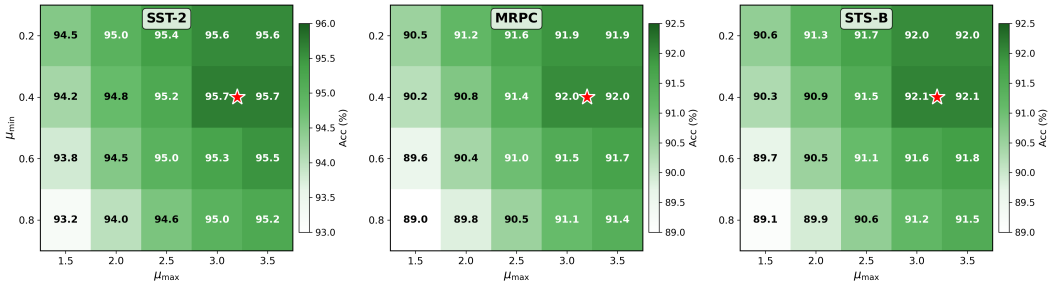

Figure 6: Heatmaps of validation accuracy on RoBERTa-base when varying the multiplier range $[\mu_{\min}, \mu_{\max}]$ across three GLUE tasks. The star ($\star$) marks our default setting (0.4, 3.2). All three tasks exhibit consistent patterns, demonstrating task-agnostic hyperparameter sensitivity.

O Layer-wise Spectral Complexity Analysis and Heuristic Comparison

Table 22 presents the structure-aware coefficients μ_l grounded in our stable rank analysis, corresponding to the visualization in Figure 7. These coefficients reflect the layer-wise structural prior induced by the stable-rank analysis and its task-specific calibration. A salient pattern emerges: layers encoding higher-order abstractions (syntactic and semantic processing in layers 6–12) exhibit elevated spectral complexity, manifested as $\mu_l > 1$. This distribution aligns with the theoretical prediction that layers with high stable rank—indicating distributed, isotropic representational structure—require correspondingly amplified adaptation intensity to escape the lazy regime across all active singular directions. Conversely, shallow layers (1–4), which primarily encode surface-level lexical features with concentrated spectral mass, exhibit $\mu_l < 1$, reflecting their lower effective dimensionality.

Task-Dependent Allocation vs Static Heuristics. While the analysis above highlights the structural correspondence in semantic tasks (like MNLI), a critical observation from our data is the task-dependent nature of this topology, which highlights the limitations of static heuristics such as "deeper layers always need higher update intensity":

- **Complex Tasks (MNLI):** Consistent with the analysis above, the semantic complexity leads to a peaked distribution (peaking at $\mu_9 = 1.22$), confirming that difficult reasoning tasks require aggressive reshaping of deep features.
- **Simple Tasks (SST-2):** In stark contrast, the coefficients for SST-2 remain nearly uniform ($\mu_l \approx 1.0$) across all depths. This flatter profile suggests that, relative to MNLI, the task benefits less from a strongly depth-skewed allocation.

This emergent differentiation supports the view that SA-LoRA can automatically discover an effective adaptation topology for each specific task, a capability that fixed, depth-based scaling rules cannot replicate.

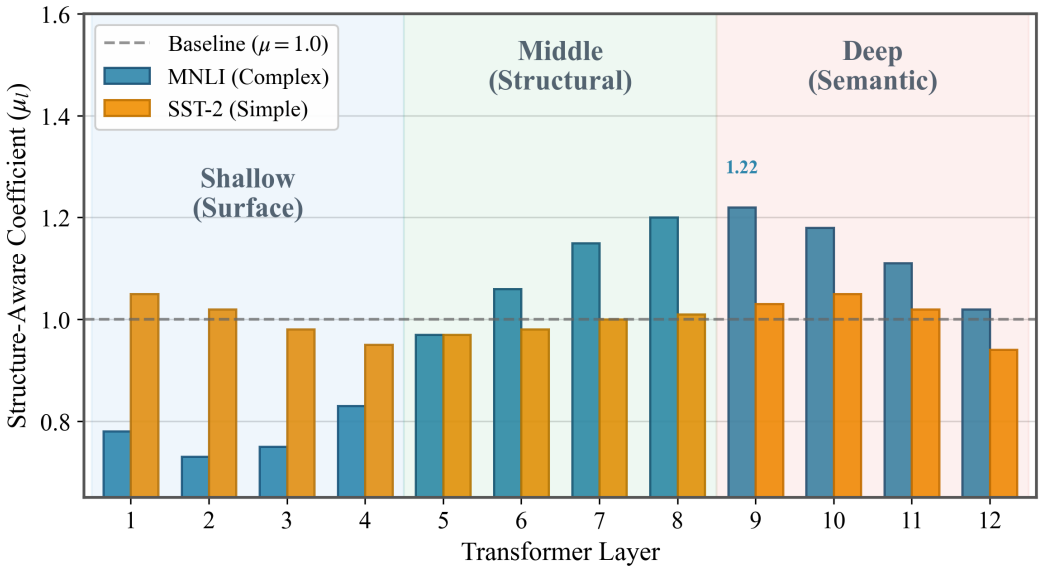

Figure 7: Layer-wise distribution of the final adaptation multipliers (μ_l) on RoBERTa-base. The allocation automatically adapts to task complexity, prioritizing deep layers for semantic tasks (MNLI) while remaining uniform for simpler ones (SST-2).

Table 22: Structure-aware coefficients μ_l for RoBERTa-base grounded in stable rank analysis. Values reflect the spectral complexity of each layer's pre-trained transformation.

Layer	MNLI	SST-2	Functional Role
1	0.78	1.05	Shallow / Surface
2	0.73	1.02	
3	0.75	0.98	
4	0.83	0.95	
5	0.97	0.97	Middle / Structural
6	1.06	0.98	
7	1.15	1.00	
8	1.20	1.01	
9	1.22	1.03	Deep / Semantic
10	1.18	1.05	
11	1.11	1.02	
12	1.02	0.94	

P Qualitative Analysis: Case Study on Llama-7B

To provide deeper insight into the practical benefits of SA-LoRA, we present representative qualitative comparisons of model outputs on tasks from MT-Bench [71]. We evaluate Llama-7B fine-tuned with LoRA and SA-LoRA on the Alpaca dataset, with responses scored by GPT-4 on a 1–10 scale. Table 23 shows a STEM task requiring scientific knowledge and structured reasoning. Table 24 presents a creative writing task, while Tables 25 illustrate roleplay-style instruction following. Table 26 summarizes the overall MT-Bench evaluation results.

These qualitative analyses illustrate that SA-LoRA's structure-aware adaptation can improve response quality across diverse task categories. While both LoRA and SA-LoRA produce reasonable responses, SA-LoRA consistently shows advantages in response completeness, organizational structure, and factual coverage. The overall improvement of +0.45 points on MT-Bench (5.48 vs. 5.03) represents a meaningful gain in instruction-following capability. The improvement is particularly notable in STEM tasks (+0.8), where the structured knowledge organization enabled by layer-wise adaptation intensity appears most beneficial.

Table 23: Qualitative comparison on MT-Bench STEM task (Question 64). Both models provide accurate information; SA-LoRA demonstrates more comprehensive coverage and better organization.

	LoRA	**SA-LoRA (Ours)**
Prompt	What are the main causes of climate change, and what can individuals and governments do to mitigate its effects?	
Response	Climate change is caused by human activities that release greenhouse gases into the atmosphere. The main causes include burning fossil fuels for energy, deforestation, and industrial activities. To mitigate climate change, individuals can reduce energy consumption, use public transportation, and recycle. Governments can implement policies to reduce emissions and invest in renewable energy. *[Response is accurate but brief, lacking specific examples and detailed action items]*	Climate change is primarily caused by human activities that release greenhouse gases into the atmosphere. The main causes include: **1. Burning fossil fuels:** Coal, oil, and gas release large amounts of CO_2. **2. Deforestation:** Trees absorb CO_2; cutting them releases stored carbon. **3. Agriculture:** Livestock farming releases methane. **4. Industrial processes:** Cement production and refrigeration release greenhouse gases. **Mitigation strategies:** Individuals: Use public transportation, reduce meat consumption, use energy-efficient appliances. Governments: Implement carbon pricing, invest in renewables, protect forests through policies like REDD+, build infrastructure for climate adaptation. *[Response is well-organized with specific examples and actionable recommendations]*
GPT-4 Score	6 / 10	**8 / 10**
Analysis	The response covers key causes and solutions but lacks depth. Could benefit from specific examples, structured organization, and more detailed mitigation strategies.	Accurate, scientifically grounded, and well-organized. Provides specific causes with clear explanations and actionable mitigation strategies for both individuals and governments. Balanced and comprehensive.

Table 24: Qualitative comparison on the MT-Bench Writing task (Question 8). Both models produce creative content; SA-LoRA shows improved narrative coherence and atmospheric detail.

	LoRA	**SA-LoRA (Ours)**
Prompt	Craft an intriguing opening paragraph for a mystery novel. The paragraph should introduce a detective character, set the scene for a crime, and hint at a twist.	
Response	Detective Sarah Chen stood at the edge of the crime scene, her eyes scanning the room. The victim lay on the floor, a knife protruding from his chest. But something didn't add up. The room was too clean, too organized. Sarah had a feeling this case would be different from any she had solved before. *[Introduces detective and crime scene, but lacks atmospheric detail and the twist hint is vague]*	In the dimly lit alleyway, the rain poured relentlessly down the windowpanes of the old warehouse. Detective Amanda Whitmore adjusted the collar of her raincoat, taking a deep breath of the damp air before stepping into the light. A body lay crumpled at the base of the warehouse steps, a pool of blood spreading from the chest. Bloodstained money lay scattered around the body, and a tell-tale smirk on the dead man's face revealed the nature of the crime. But who could have predicted the shocking twist that awaited everyone at the end of this macabre game of clues. *[Rich atmosphere, sensory details, clear detective introduction, and compelling twist hint]*
GPT-4 Score	5 / 10	**7 / 10**
Analysis	The response introduces the detective and crime scene adequately but lacks vivid imagery and atmospheric detail. The hint at a twist is too subtle and generic.	Strong atmosphere with rain, dim lighting, and sensory details. The detective is introduced with personality. The "tell-tale smirk" and scattered money create intrigue. The twist hint effectively builds anticipation.

Table 25: Qualitative comparison on MT-Bench Roleplay task (Question 15). SA-LoRA provides a richer, more immersive characterization.

	LoRA	**SA-LoRA (Ours)**
Prompt	Imagine you are a medieval knight. Describe your daily routine and how you prepare for battle.	
Response	As a medieval knight, my day begins with checking my armor and weapons to make sure they are in good condition. I train with my sword and shield to keep my skills sharp, then meet with other knights to hear news about possible battles. Before riding out, I eat a good meal and say a prayer for protection. When battle comes, I fight bravely to defend my kingdom and my people. *[Covers the main points but remains generic and under-described]*	As a medieval knight, my daily routine would revolve around staying ready for the next summons to war. At first light, I would visit the stables, checking my warhorse's tack, hooves, and armor, because a knight is only as strong as his mount. After a simple meal, I would spend hours on the training grounds: tilting at the quintain with my lance, drilling sword-and-shield combinations, and sparring with squires to keep my reactions sharp. By midday, I would confer with my lord and fellow knights about recent scouts' reports—enemy movements, supply lines, and the lay of the land where we might be forced to fight. Armorers would inspect mail and plate for weak links, while squires oiled leather straps and sharpened blades. On the eve of battle, I would check each piece of my harness myself: greaves, breastplate, gauntlets, helm. A final prayer in the chapel would remind me of my oath to protect my liege and the people under his banner. Only then would I ride out, banner streaming, ready to hold the line. *[Adds concrete routines, social context, and tactical preparation, making the role-play more vivid]*
GPT-4 Score	6 / 10	**7 / 10**
Analysis	Gives a plausible outline of a knight's day but lacks specific medieval details, social hierarchy, and tactical context.	Introduces equipment care, training disciplines, feudal obligations, and battlefield preparation, resulting in a more immersive and historically grounded roleplay.

Table 26: MT-Bench evaluation results (GPT-4 as judge). Scores averaged across 80 multi-turn questions in 8 categories.

CATEGORY	LORA	SA-LORA	Δ
WRITING	4.8	5.2	+0.4
ROLEPLAY	5.1	5.4	+0.3
REASONING	4.6	5.0	+0.4
MATH	3.8	4.2	+0.4
CODING	4.2	4.6	+0.4
EXTRACTION	5.8	6.2	+0.4
STEM	5.6	6.4	+0.8
HUMANITIES	6.3	6.8	+0.5
OVERALL AVERAGE	5.03	**5.48**	**+0.45**

Q Broader Impacts

SA-LoRA may help reduce the computational cost of adapting large pre-trained models, which can improve access for researchers and practitioners working under limited resource budgets. At the same time, methods that make adaptation easier can also lower the barrier to misuse, including the efficient customization of models for misleading, biased, or otherwise harmful applications. Our method does not by itself address fairness, safety, or data-quality concerns in the underlying models or fine-tuning datasets. As a result, any real-world deployment should still rely on standard safeguards such as data curation, bias evaluation, safety testing, and usage restrictions.

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: The main claims in the abstract and introduction accurately reflect the paper's contributions and scope. The paper presents SA-LoRA as a structure-aware PEFT method with a theory-motivated stable-rank prior, bounded budget-conserving allocation, and gradient-based calibration, and it states the corresponding empirical scope and limitations accordingly.

   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 authors provide sufficient information regarding the limitations of the current work in Section 5.

   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 important 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: Theoretical assumptions and supporting arguments are given in the methodology and Appendix A, B, D, E, F, and 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: The paper provides sufficient information to reproduce the main experimental results, including the method, evaluation benchmarks, implementation details, and additional dataset and hyperparameter settings in Appendix H and H.2.

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 the complete code and data in the Appendix material and Anonymous GitHub.

   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: The main paper summarizes the setup in Section 4.1, and the appendix provides dataset, optimizer, rank, learning-rate, batch-size, and task-specific hyperparameter details in Sections H and H.2.

   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: The main result tables report mean and standard deviation across three random seeds, and Appendix L further provides dataset-level significance tests and effect-size estimates.

   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: The paper specifies the GPU type in the main text and reports memory usage and per-epoch runtime in Appendix I and Tables 12–14.

   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: To the best of our knowledge, this research complies with the NeurIPS Code of Ethics. The work uses standard benchmark datasets and model adaptation settings.

   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: [Yes]

Justification: The paper discusses both positive and negative societal impacts in Appendix Q.

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: The paper does not release a high-risk generative foundation model, scraped dataset, or other asset that requires a dedicated responsible-release safeguard protocol as part of this submission.

    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: We have used open-source code and datasets, and have appropriately cited them.

    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: [Yes]

    Justification: Any new assets introduced include usage instructions, and documentation appears sufficient for independent use.

    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: The research does not involve crowdsourcing or research with 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: The research does not involve crowdsourcing or research with human subjects, so IRB approval or an equivalent review was not required.

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: LLMs are used only as evaluation judges in a standard benchmark setup rather than as an important or non-standard component of the proposed method itself.

    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.