Navigating the Optimization Landscape of Decoupled Learning Rates in LoRA-based methods

Anonymous ACL submission

Abstract

Parameter-efficient fine-tuning (PEFT) methods such as LoRA enable efficient adaptation of large language models by updating only a small subset of parameters, but typically rely on a single learning rate for all adapter components. Recent work suggests that decoupling learning rates across low-rank factors can improve optimization by accounting for asymmetric gradient scales. In this paper, we systematically study the optimization landscape induced by decoupled learning rates in PEFT methods, including LoRA, VeRA, and AdaLoRA. Using extensive hyperparameter optimization with Optuna's TPE sampler on GLUE benchmarks, we show that decoupling introduces pronounced sensitivity to learning rate configurations, leading to unstable training dynamics and high variance in performance across tasks and methods. We characterize these effects and identify consistent patterns governing interactions between learning rates of different parameter groups. Based on our findings, we derive practical guidelines for robust hyperparameter selection and highlight limitations of standard optimization strategies in this setting.

1 Introduction

Recent advances in large language models (LLMs) Vaswani et al. (2023); Devlin et al. (2019); Brown et al. (2020) have been driven largely by increases in model scale, both in terms of parameter counts and training data. However, these improvements now exhibit diminishing returns (Kaplan et al., 2020). As models continue to grow, training costs rise dramatically, placing full end-to-end training beyond the reach of most researchers and limiting many studies to inference-only settings (Min et al., 2022; Dai et al., 2023; Xie et al., 2022). This trend concentrates cutting-edge model development within organizations with substantial computational resources, thereby constraining broader scientific participation.

In contrast, optimization strategies for the learning process itself—rather than architectural innovations—have historically received comparatively less attention. The escalating cost of large-scale training has renewed interest in more efficient learning paradigms, leading to rapid progress in parameter-efficient training methods. Seminal work by Houlsby et al. (2019) and Hu et al. (2021) demonstrated that lightweight, trainable components can adapt large pretrained models with performance competitive with, and in some cases surpassing, full fine-tuning. These approaches enable specialization of modern LLMs using modest hardware resources—often comparable to those required for inference alone—and significantly smaller task-specific datasets when leveraging pretrained representations.

A recent line of work has shown that even within these lightweight adapters, the choice of optimizer hyperparameters—particularly the learning-rate assignment to the low-rank matrices—can have an outsized impact on downstream performance (Hayou et al., 2024). However, the precise mechanisms, stability, and transferability of such learning-rate decoupling remain underexplored beyond the original LoRA setting.

2 Our contributions

In this paper we conduct a systematic empirical study of learning-rate decoupling in parameter-efficient fine-tuning. Using efficient hyperparameter search in an NLP setting, we analyze its effects on validation metrics for LoRA (Hu et al., 2021) and its optimizer-level variant LoRA+ (Hayou et al., 2024). We pay particular attention to the stability of the observed gains and the structure of the hyperparameter search space.

Building on this analysis, we introduce **VeRA+** and **AdaLoRA+**, which apply the same asymmetric learning-rate strategy to the highly parameter-

efficient VeRA adapter (Kopiczko et al., 2024) and AdaLoRA (Zhang et al., 2023).

Our experiments on RoBERTa-base and RoBERTa-large across a representative subset of the GLUE benchmark show that both proposed methods consistently outperform their standard counterparts in the tested regime. These results provide initial evidence that optimizer-level refinements such as learning-rate decoupling can generalize beyond standard LoRA to other members of the LoRA family, even under extremely low parameter budgets. We discuss the scope and limitations of these findings with respect to model scale in Section 7.

3 Related Work

Parameter-efficient fine-tuning (PEFT) has become essential for adapting large language models under constrained computational resources. Early approaches, such as adapter-based methods (Houlsby et al., 2019) and Low-Rank Adaptation (Hu et al., 2021), demonstrated that small, dedicated modules or low-rank updates can match or exceed full fine-tuning performance while updating only a tiny fraction of parameters.

3.1 Foundational Large Language Models

The development of PEFT builds on the success of foundational large language models. Early autoregressive models like GPT introduced generative pre-training for broad language understanding (Radford et al., 2018). Bidirectional encoder-only architectures followed with BERT, which uses masked language modelling to capture rich contextual representations (Devlin et al., 2019). RoBERTa, the base model employed in this work, builds directly on BERT by optimizing key design choices—such as removing the next-sentence prediction objective, using larger batches, more data, and dynamic masking—leading to substantially improved robustness and downstream performance (Liu et al., 2019). Subsequent scaling efforts produced massive models like GPT-3, demonstrating remarkable few-shot and zero-shot generalization through sheer parameter count and data volume (Brown et al., 2020).

3.2 Hyperparameter Optimization

Effective adaptation of these large models requires careful hyperparameter tuning, especially given the high computational cost of fine-tuning. Hyperparameter optimization (HPO) is therefore critical for maximizing performance in resource-intensive LLM fine-tuning scenarios. Bayesian optimization has emerged as an efficient approach for navigating high-dimensional, expensive-to-evaluate search spaces in machine learning (Snoek et al., 2012). Modern frameworks build on these ideas with practical, scalable implementations; Optuna, used in this study, introduces a define-by-run API, efficient pruning, and flexible samplers (including TPE and CMA-ES) to accelerate the search process while supporting distributed and multi-objective tuning (Akiba et al., 2019).

3.3 Parameter-Efficient Fine-Tuning for LLMs

Building on these foundations, research has significantly expanded the PEFT design space beyond initial methods. Quantization-aware techniques like LoftQ (Li et al., 2023) and QLoRA (Dettmers et al., 2023) further improve memory efficiency by combining low-precision representations with low-rank adapters. Variants building directly on LoRA include adaptive rank allocation, for example, AdaLoRA (Zhang et al., 2023), and reparameterization improvements, such as DoRA (Liu et al., 2024), which decouples magnitude and direction for better optimization stability.

Other directions explore meta-learning-inspired adapters for improved generalization and data efficiency (Topal et al., 2025; Cheng et al., 2025), expert-based architectures like Mixture of Dyadic Experts to enhance multitask capacity (Ning et al., 2024), and ultra-efficient formulations such as VeRA (Kopiczko et al., 2024), which uses random frozen matrices with trainable scaling vectors. Recent surveys provide comprehensive taxonomies of PEFT methods, categorizing them into additive, reparameterization-based, selective, hybrid, and quantization-aware families, while highlighting trade-offs in efficiency, performance, and applicability (Han et al., 2024; Lialin et al., 2024).

Optimizer-level refinements have also shown promise within this evolving landscape. In particular, LoRA+ (Hayou et al., 2024) demonstrates that asymmetric learning rates for the low-rank matrices yield consistent gains and faster convergence compared to standard LoRA. This motivates further exploration of decoupling strategies in related adapters, including highly parameter-efficient variants such as VeRA.

4 Method

Our approaches combine two previously introduced techniques: VeRA (Kopiczko et al., 2024) and AdaLoRA (Zhang et al., 2023) with LoRA+ (Hayou et al., 2024). While the method is model-agnostic, we focus on its application to large language models in natural language processing tasks, similarly to the cited works.

We begin with the standard LoRA formulation (Hu et al., 2021), in which a pretrained weight matrix $W_0 \in \mathbb{R}^{m\times n}$ is adapted by constraining its update ΔW to a low-rank decomposition. The forward pass is then written as

$$h = W_0 x + \Delta W x = W_0 x + \underline{BA}\, x, \qquad (1)$$

where the underlined components denote trainable parameters. The matrices $A \in \mathbb{R}^{r\times m}$ and $B \in \mathbb{R}^{n\times r}$ have rank $r \ll \min(m, n)$, allowing LoRA to freeze W_0 while updating only a small number of additional parameters.

VeRA (Kopiczko et al., 2024) adopts a similar low-rank structure but replaces A and B with frozen, random, shared matrices while introducing lightweight trainable scaling vectors. The VeRA update is expressed as

$$h = W_0 x + \Delta W x = W_0 x + \underline{\Lambda_b}\, B\, \underline{\Lambda_a}\, A x, \quad (2)$$

where A and B are fixed, and Λ_a and Λ_b are diagonal matrices containing the trainable scaling vectors. This parameterization allows VeRA to modulate rows and columns of the low-rank factors while keeping the number of learnable parameters extremely small.

AdaLoRA (Zhang et al., 2023) focuses on more efficient budget distribution between layers rather than further optimizing the low-rank representation itself. This enables the model to maintain the same total number of trainable parameters while allocating larger budgets to layers that require more complex adjustments. The approach is achieved using singular value decomposition (SVD). The update is expressed as

$$h = W_0 x + \Delta W x = W_0 x + \underline{P\Lambda Q}, \qquad (3)$$

where $P \in \mathbb{R}^{m\times r}$ and $Q \in \mathbb{R}^{r\times n}$ are orthonormal factors and Λ is a diagonal matrix of singular values.

The underlined parameters (P, Q, and Λ) are trained while the orthonormality conditions are enforced via regularization. Thanks to this formulation, a global rank allocator can determine—based on the learned singular values in Λ—which layers can operate with a smaller effective rank while keeping the overall parameter budget constant.

A common characteristic of LoRA-style methods, including VeRA and AdaLoRA, is that they typically rely on a single learning rate for all trainable components. However, LoRA+ (Hayou et al., 2024) challenges this assumption. Using scaling arguments in the infinite-width regime, the authors show that employing a unified learning rate becomes increasingly suboptimal as model width grows—a regime in which modern LLMs clearly operate. Their analysis reveals that the gradients of the low-rank factors A and B exhibit systematically different magnitudes during training. Consequently, they argue that fine-tuning benefits from a learning-rate ratio satisfying:

$$\eta_B/\eta_A \gg 1.$$

Motivated by these findings, we extend the learning-rate decoupling principle to VeRA and AdaLoRA, introducing VeRA+ and AdaLoRA+. In VeRA+, we assign separate learning rates η_a and η_b (also referred to as lr_a/lr_b) to the trainable scaling vectors Λ_a and Λ_b, respectively. In AdaLoRA+, we use separate learning rates for the trainable components P, Q, and Λ, with P and Λ configured to share the same learning rate. From this point onward, the learning rates are referred to as η_a/η_b or lr_a/lr_b. Our hypothesis is that, just as in LoRA+, the asymmetry in gradient scales persists in VeRA's and AdaLoRA's parameterizations, and that learning-rate decoupling can improve adaptation quality, convergence speed, or stability without increasing the parameter count. We empirically investigate this hypothesis using RoBERTa-base and RoBERTa-large across a representative subset of the GLUE benchmark.

5 Experiments

We evaluate the proposed VeRA+ and AdaLoRA+ methods as well as LoRA+, along with their standard counterparts, on a representative subset of the GLUE benchmark (Wang et al., 2019). This subset includes the following tasks: CoLA, MRPC, RTE, SST-2, and STS-B. We omit MNLI, QNLI, QQP, and WNLI due to computational constraints.

For each task, model, and method, we perform an independent hyperparameter optimization (HPO) using Optuna (Akiba et al., 2019) with the Tree-structured Parzen Estimator (TPE) sampler. This per-method optimization ensures a fair comparison by allowing each variant (standard vs. decoupled) to use its individually optimized hyperparameters. It also enables us to analyze the structure and behavior of the hyperparameter search space.

5.1 Models

We experiment with two widely used pretrained Transformer models: RoBERTa-base (125M parameters) and RoBERTa-large (355M parameters) (Liu et al., 2019). This choice allows us to examine the impact of model scale on learning-rate decoupling and to investigate whether the scaling trends reported in LoRA+—where wider networks benefit more from asymmetric learning rates—also appear in the parameterizations of VeRA and AdaLoRA.

5.2 Parameter-Efficient Fine-Tuning Setup

For all LoRA-, VeRA-, and AdaLoRA-based methods, adapters are inserted into the query and value projection matrices of every Transformer layer. For standard LoRA and VeRA, we fix the rank to $r = 8$. For AdaLoRA, we set the initial rank to 12 and the target rank to 8 to enable its dynamic rank allocation mechanism. Learning-rate decoupling is applied where relevant. We denote the base learning rate by η and the learning-rate ratio by $\rho = \eta_B/\eta_A$.

5.3 Hyperparameter Optimization

Our HPO focuses on the effects of learning-rate decoupling. We consider two experimental settings for each method:

- **Variable-ratio search** (LoRA+, VeRA+, AdaLoRA+): a full sweep over both the base learning rate η and the ratio ρ.
- **Ratio-1.0 baseline** (standard LoRA, VeRA, AdaLoRA): a sweep over η only, with the ratio fixed at $\rho = 1.0$.

All searches use Optuna with the TPE sampler and are performed independently for each task–model pair. The initial search space was

$$\eta \in [10^{-8}, 10^{-2}], \quad \rho \in [10^{-5}, 10^{5}],$$

with 100 trials per study.

Preliminary results revealed that optimal configurations often lay outside parts of this broad space. We therefore conducted follow-up searches with two refined spaces informed by the initial outcomes:

$$\eta \in [10^{-3}, 10^{-2}], \quad \rho \in [10^{-5}, 1]$$

and

$$\eta \in [10^{-5}, 5 \times 10^{-3}], \quad \rho \in [1, 10^{2}].$$

5.4 Training Protocol

All trials use a batch size of 32, the AdamW optimizer, and a linear learning-rate scheduler with a warmup ratio of 0.1. Models are trained for a fixed number of epochs per task (10 epochs for CoLA, MRPC, STS-B, RTE, and SST-2).

During HPO, we enable Optuna's pruning mechanism to reduce computational cost. Intermediate evaluations occur every 500 training steps after an initial warmup of 1,500 steps. After identifying the best hyperparameter configuration for each task and method, we retrain the model from scratch using the optimal settings for the same number of epochs (with pruning disabled) to obtain stable final performance.

5.5 Evaluation Metrics

During HPO, model selection relies on task-specific validation metrics: accuracy for SST-2, RTE, and STS-B; Matthews correlation coefficient for CoLA; and F1 score for MRPC. Final results are averaged over five independent runs with different random seeds, reporting mean and standard deviation.

5.6 Hardware

All experiments were conducted on a machine equipped with two RTX 5070 Ti GPUs, with multiple experiments running in parallel across the GPUs to reduce total wall-clock time.

6 Results

We present the main findings from our hyperparameter optimization study and performance evaluation of standard and decoupled variants (LoRA+/VeRA+/AdaLoRA+) across the selected GLUE tasks. A comprehensive presentation of the results is provided in Appendix Section B, with condensed summaries reported in Tables 1 and 2. The corresponding search-space visualizations are depicted in Figures 1.

Overall, the benefits of learning-rate decoupling ($\rho \neq 1$) are task- and adapter-dependent rather than universally following the LoRA+ recommendation of $\rho \gg 1$. In many configurations, the optimal ratio satisfies $\rho \leq 1$, with smaller values of ρ often yielding competitive or superior validation performance compared to the symmetric baseline ($\rho = 1$).

For AdaLoRA, the behavior closely resembles that of standard LoRA. The search-space visualizations reveal a similar structure, albeit somewhat contracted along the lr_A axis. The optimal configurations predominantly favor $\rho \leq 1$ for both RoBERTa-base and RoBERTa-large. The LoRA+ guideline of strongly asymmetric rates ($\rho \gg 1$) does not consistently emerge in the tested regime.

VeRA exhibits the most distinct pattern. The search results indicate that lr_B (corresponding to the second scaling vector) often plays a minimal role, with performance being driven primarily by lr_A. Consequently, the optimal ratio is frequently close to $\rho \approx 0$. This suggests that, for most tasks in our setup, one scaling vector dominates the adaptation process in VeRA, although exceptions exist where lr_B contributes meaningfully. Despite this asymmetry in sensitivity, VeRA+ still achieves competitive performance without increasing the parameter budget.

Task-specific differences are evident in the hyperparameter landscapes. Tasks such as CoLA and especially RTE display highly constrained regions of high-performing configurations, whereas others (e.g., SST-2, MRPC) tolerate wider ranges of effective learning rates and ratios. These variations do not appear to correlate solely with dataset size; task difficulty and the nature of the required adaptation likely play important roles.

Regarding model scale, RoBERTa-large shows a modest shift toward higher optimal ρ values compared to RoBERTa-base in several cases, providing weak support for the idea that wider networks may benefit more from decoupling. However, with only two model scales, it remains difficult to extract general scaling rules. A more comprehensive study across additional model sizes would be needed to confirm or refute stronger trends.

In terms of final performance (Table 1), the decoupled variants achieve the best result in many but not all task–model combinations. Gains are generally modest and consistent with the observation that optimal decoupling is often mild ($\rho \lesssim 1$) rather than extreme. The $\rho \leq 1$ regime frequently outperforms both the $\rho \geq 1$ regime and the symmetric baseline, highlighting that the direction and magnitude of beneficial asymmetry depend on the specific adapter parameterization.

7 Discussion

Our empirical study shows that learning-rate decoupling consistently yields modest improvements over unified learning rates across LoRA, VeRA, and AdaLoRA variants, without incurring any additional parameter or memory overhead. In several configurations the gains are more noticeable; for instance, AdaLoRA+ on RoBERTa-large improves the MRPC score by 0.008 over the symmetric baseline. While these improvements are generally small, they demonstrate that simple optimizer-level changes can still provide measurable benefits in extremely low-parameter regimes.

The effectiveness of decoupling is highly task- and adapter-dependent. For standard LoRA and AdaLoRA, our results partially align with the scaling arguments in LoRA+ (Hayou et al., 2024): optimal ratios frequently deviate from $\rho = 1$, and RoBERTa-large shows a weak tendency toward higher ρ values compared to the base model. However, the strong asymmetry ($\rho \gg 1$) advocated by LoRA+ does not consistently materialize in our experiments. This suggests that the practical benefit of decoupling depends on model width, task characteristics, and the specific adapter parameterization more than previously emphasized.

VeRA exhibits the most distinctive pattern. Performance is often dominated by a single scaling vector, pushing optimal ratios toward the extremes ($\rho \approx 0$ or, less frequently, very large values). We attribute this behavior to the constrained nature of VeRA's design. While LoRA and AdaLoRA allow the optimizer to adjust both direction and magnitude through full low-rank matrices (or their SVD decomposition), VeRA only permits independent scaling along two fixed random transformation matrices. This limited degree of freedom appears to force the hyperparameter optimizer to heavily emphasize one scaling vector over the other for most tasks.

This finding raises a natural question: whether increasing the rank r would mitigate the dominance of a single vector by reducing the impact of randomness in the frozen matrices. We also observe that some optimal VeRA configurations fall into regimes where AdamW can become unstable; in such cases, additional training epochs or more con-

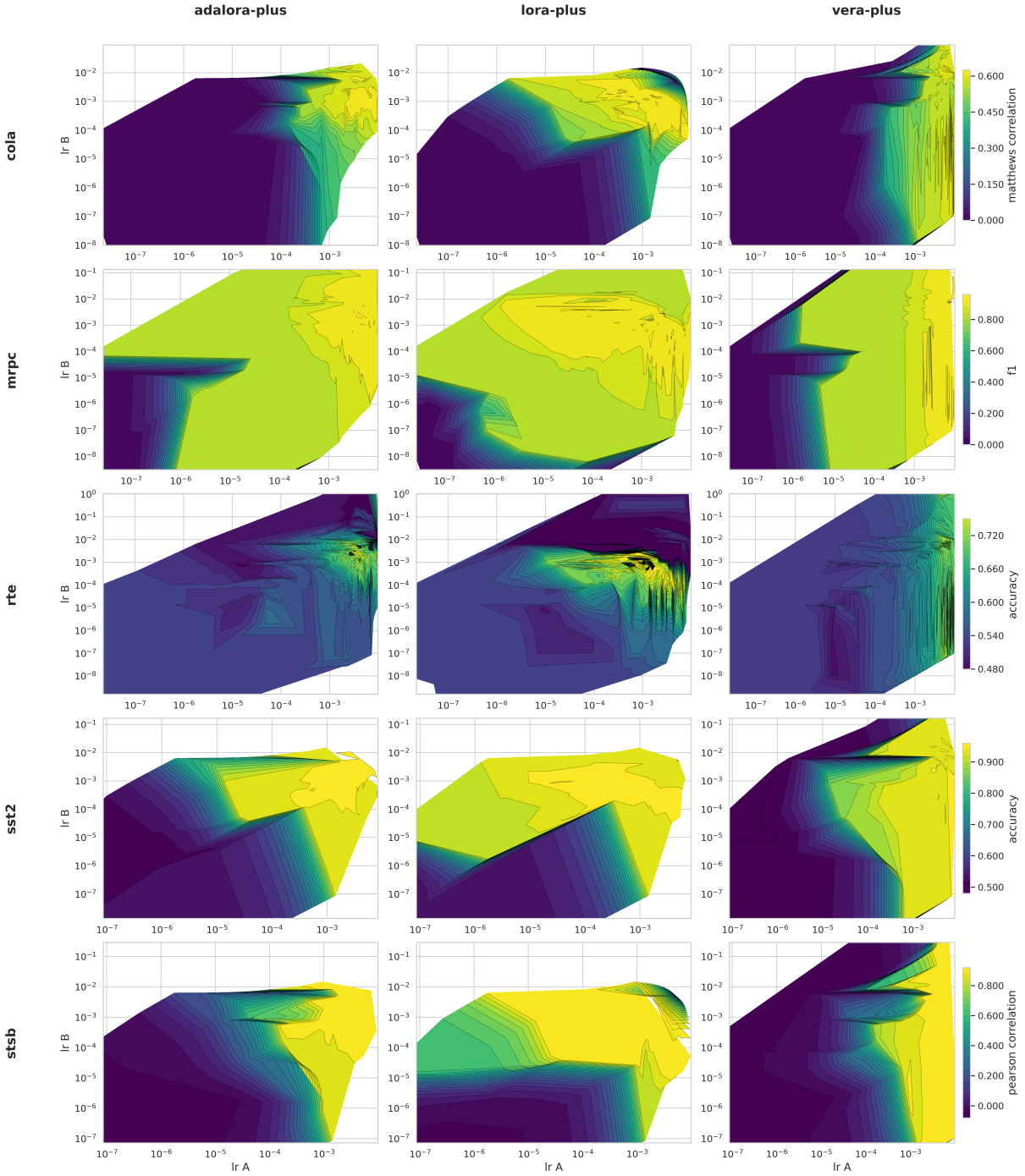

Figure 1: Search space visualization Roberta-base across different adapters and tasks. Created based on all finished trials from all of the search spaces combined.

Table 1: Performance comparison of different models, adapters, and search types across GLUE tasks. Bold indicates the best value.

Model	Adapter	ρ Range	CoLA	MRPC	RTE	SST-2	STS-B
RoBERTa base	AdaLoRA	full	**0.6131**	0.9117	**0.7307**	0.9433	**0.9066**
	AdaLoRA+	$\rho \geq 1$	0.6017	0.9113	0.6809	0.9445	0.9032
		$\rho \leq 1$	0.6125	0.9091	0.7069	0.9450	0.9058
		$\rho = 1$	0.6036	**0.9121**	0.7256	**0.9463**	0.9051
	LoRA	full	0.6176	**0.9194**	0.7415	**0.9463**	**0.9081**
	LoRA+	$\rho \geq 1$	0.6095	0.9143	0.6968	0.9450	0.9029
		$\rho \leq 1$	0.6194	0.9181	**0.7574**	0.9447	0.9066
		$\rho = 1$	**0.6209**	0.9119	0.7567	0.9443	0.9072
	VeRA	full	0.5832	**0.9124**	**0.7054**	**0.9397**	0.8983
	VeRA+	$\rho \geq 1$	**0.5835**	0.9070	0.6931	0.9388	0.8984
		$\rho \leq 1$	0.5653	0.9038	0.6838	0.9369	0.8922
		$\rho = 1$	0.5809	0.9097	0.7004	0.9383	**0.8987**
RoBERTa large	AdaLoRA	full	0.6425	0.8359	0.8079	0.9603	0.9156
	AdaLoRA+	$\rho \geq 1$	0.6411	**0.9195**	0.8079	0.9601	0.9159
		$\rho \leq 1$	**0.6493**	0.9167	0.8087	**0.9606**	**0.9176**
		$\rho = 1$	0.6388	0.8352	**0.8137**	**0.9606**	0.9145
	LoRA	full	0.5097	0.9012	0.8383	0.9624	**0.9209**
	LoRA+	$\rho \geq 1$	0.6430	0.8971	0.8289	**0.9626**	0.9205
		$\rho \leq 1$	**0.6564**	**0.9203**	0.8397	0.9622	0.9194
		$\rho = 1$	0.6549	0.9043	**0.8419**	0.9622	0.9206
	VeRA	full	0.6310	0.9085	0.8051	0.9596	0.9100
	VeRA+	$\rho \geq 1$	0.6287	0.9178	0.7935	**0.9610**	**0.9121**
		$\rho \leq 1$	0.6113	**0.9183**	**0.8079**	0.9601	0.9054
		$\rho = 1$	**0.6450**	0.9131	0.8022	0.9585	0.9101

Table 2: Ratio of learning rates ρ across different models, adapters, and search types on GLUE tasks. Bold indicates configuration with highest metric result.

Model	Adapter	ρ Range	CoLA	MRPC	RTE	SST-2	STS-B
RoBERTa base	AdaLoRA	full	**0.28**	0.60	**0.56**	0.41	**0.33**
	AdaLoRA+	$\rho \geq 1$	1.08	1.01	1.02	1.38	1.57
		$\rho \leq 1$	0.15	0.18	0.25	0.57	0.47
		$\rho = 1$	1.00	**1.00**	1.00	**1.00**	1.00
	LoRA	full	2.24	**0.24**	0.03	**1.21**	**0.18**
	LoRA+	$\rho \geq 1$	5.79	4.03	1.68	10.80	9.77
		$\rho \leq 1$	0.05	0.12	**0.46**	0.45	0.47
		$\rho = 1$	**1.00**	1.00	1.00	1.00	1.00
	VeRA	full	0.00	**0.00**	**0.00**	**0.01**	33.22
	VeRA+	$\rho \geq 1$	**3.05**	1.34	1.39	6.08	1.12
		$\rho \leq 1$	0.20	0.00	0.02	0.01	0.00
		$\rho = 1$	1.00	1.00	1.00	1.00	**1.00**
RoBERTa large	AdaLoRA	full	0.36	0.38	0.40	0.42	1.35
	AdaLoRA+	$\rho \geq 1$	1.85	**2.30**	1.33	1.06	1.00
		$\rho \leq 1$	**0.47**	0.56	0.44	**0.11**	**0.38**
		$\rho = 1$	1.00	1.00	**1.00**	**1.00**	1.00
	LoRA	full	21.72	0.66	0.00	19.59	**2.44**
	LoRA+	$\rho \geq 1$	19.29	3.08	6.08	**4.96**	4.41
		$\rho \leq 1$	**0.01**	**0.04**	0.04	0.17	0.05
		$\rho = 1$	1.00	1.00	**1.00**	1.00	1.00
	VeRA	full	0.01	0.00	7,543.86	0.00	0.14
	VeRA+	$\rho \geq 1$	4.17	13.09	1.39	**8.30**	**2.05**
		$\rho \leq 1$	0.00	**0.00**	**0.00**	0.21	0.00
		$\rho = 1$	**1.00**	1.00	1.00	1.00	1.00

servative scheduling may be beneficial.

From a practical standpoint, we offer the following guidelines for applying learning-rate decoupling to LoRA-family methods:

- For most adapters (LoRA and AdaLoRA), a modest joint search over the base learning rate η and the ratio ρ is worthwhile and computationally cheap.

- For highly constrained adapters such as VeRA, more careful tuning of ρ is recommended to avoid numerical instability or extreme weight values.

Our study is limited to RoBERTa-base and RoBERTa-large. Although we observe a mild shift toward higher optimal ρ with increased model scale, drawing strong scaling conclusions requires experiments on significantly larger models. Future work could examine the transferability of these findings to decoder-only models, instruction tuning, or multimodal settings, and develop theoretical analyses of gradient dynamics in constrained adapters like VeRA.

8 Conclusion

This paper investigated the generalization of learning-rate decoupling beyond standard LoRA to two other members of the LoRA family: VeRA and AdaLoRA. Through extensive hyperparameter optimization on RoBERTa-base and RoBERTa-large across a subset of the GLUE benchmark, we showed that asymmetric learning rates can provide consistent, low-cost improvements even in ultra-low-parameter regimes.

Our analysis reveals that the benefits of decoupling are adapter- and task-dependent. While LoRA+ style strong asymmetry ($\rho \gg 1$) does not always hold, milder decoupling ($\rho \lesssim 1$) frequently outperforms the symmetric baseline. VeRA exhibits particularly strong sensitivity to one scaling vector, highlighting how the degree of freedom in an adapter's parameterization influences optimal hyperparameter choices.

These results reinforce that optimizer-level refinements remain a promising direction for improving parameter-efficient fine-tuning. By requiring no extra parameters or memory, such techniques help democratize the adaptation of large language models for researchers with limited computational resources.

Future work includes scaling the study to larger models, exploring theoretical explanations for the observed asymmetries in constrained adapters, and testing the approach on a broader range of tasks and architectures.

Limitations

While our results suggest that learning-rate decoupling can provide consistent gains across several LoRA-family methods, this study has several limitations.

First, our experiments are restricted to encoder-only models (RoBERTa-base and RoBERTa-large) and relatively moderate model scales. Although we observe weak indications that optimal decoupling behavior may shift with increasing width, these results do not establish scaling laws comparable to those hypothesized in prior work. It remains unclear whether the observed trends transfer to much larger foundation models, particularly decoder-only LLMs used in contemporary instruction tuning.

Second, the empirical evaluation is limited to a subset of GLUE tasks. While these tasks cover diverse linguistic phenomena, they represent relatively small-scale supervised classification and regression problems. The conclusions may not generalize to generative tasks, long-context reasoning, multilingual adaptation, or domains where PEFT methods are commonly deployed today.

Third, despite extensive hyperparameter optimization, our conclusions are necessarily conditioned on the explored search spaces, optimization budget, and the use of TPE-based search. Alternative samplers, larger search budgets, or different parameterizations of the decoupled learning rates could potentially reveal different optima or reduce some of the instability we observed.

Fourth, our study focuses primarily on empirical optimization behavior and does not provide a theoretical explanation for why decoupling behaves differently across LoRA, VeRA, and AdaLoRA. In particular, the strong asymmetries observed in VeRA remain only partially understood and warrant deeper analysis of gradient dynamics and parameter interactions.

Fifth, although decoupling introduces no additional trainable parameters, it increases hyperparameter tuning complexity. In practice, some reported gains are modest and may depend on access to nontrivial HPO resources, which could reduce the practical benefit in highly resource-constrained

settings.

Finally, our proposed extensions VeRA+ and AdaLoRA+ are evaluated under a fixed set of architectural and training choices (rank settings, optimizer, scheduler, adapter placement). Different design choices may interact with learning-rate decoupling in ways not captured in this study.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. *Preprint*, arXiv:1907.10902.

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, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Bo Cheng, Xu Wang, Jinda Liu, Yi Chang, and Yuan Wu. 2025. Meta-lora: Data-efficient multi-task fine-tuning for large language models. *Preprint*, arXiv:2510.11598.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. *Preprint*, arXiv:2212.10559.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

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

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

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *Preprint*, arXiv:1902.00751.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

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

Dawid J. Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. 2024. Vera: Vector-based random matrix adaptation. *Preprint*, arXiv:2310.11454.

Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. 2023. Loftq: Lora-fine-tuning-aware quantization for large language models. *Preprint*, arXiv:2310.08659.

Vladislav Lialin, Vijeta Deshpande, Xiaowei Yao, and Anna Rumshisky. 2024. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *Preprint*, arXiv:2303.15647.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. Dora: Weight-decomposed low-rank adaptation. *Preprint*, arXiv:2402.09353.

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

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *Preprint*, arXiv:2202.12837.

Lin Ning, Harsh Lara, Meiqi Guo, and Abhinav Rastogi. 2024. Mode: Effective multi-task parameter efficient fine-tuning with a mixture of dyadic experts. *Preprint*, arXiv:2408.01505.

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

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical bayesian optimization of machine learning algorithms. *Preprint*, arXiv:1206.2944.

Barış Batuhan Topal, Umut Özyurt, Zafer Doğan Budak, and Ramazan Gokberk Cinbis. 2025. Meta-lora: Meta-learning lora components for domain-aware id personalization. *Preprint*, arXiv:2503.22352.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Preprint*, arXiv:1804.07461.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. *Preprint*, arXiv:2111.02080.

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

A Additional Search Space Visualization for RoBERTa-Large

Figure 2 provides the complete search-space visualization for RoBERTa-large. We omitted this figure from the main paper because its structure closely mirrors the RoBERTa-base visualization shown in Figure 1 and does not reveal qualitatively new patterns beyond those already discussed in Section 7. We include it here for completeness and transparency, allowing readers to verify that the trends observed for the base model largely persist at the larger scale.

B Detailed Learning Rate Search Results

The main paper reports condensed summaries of performance and optimal learning-rate ratios (Tables 1 and 2), while omitting the full per-task learning-rate configurations for space reasons. We provide those detailed search outcomes here in the interest of transparency and reproducibility.

These tables report the best learning-rate assignments discovered during hyperparameter optimization for each adapter, task, and ratio regime. While not necessary for interpreting the main conclusions, they may be useful for reproducing experiments, examining optimization trends at a finer granularity, or informing future studies on decoupled learning-rate parameterizations.

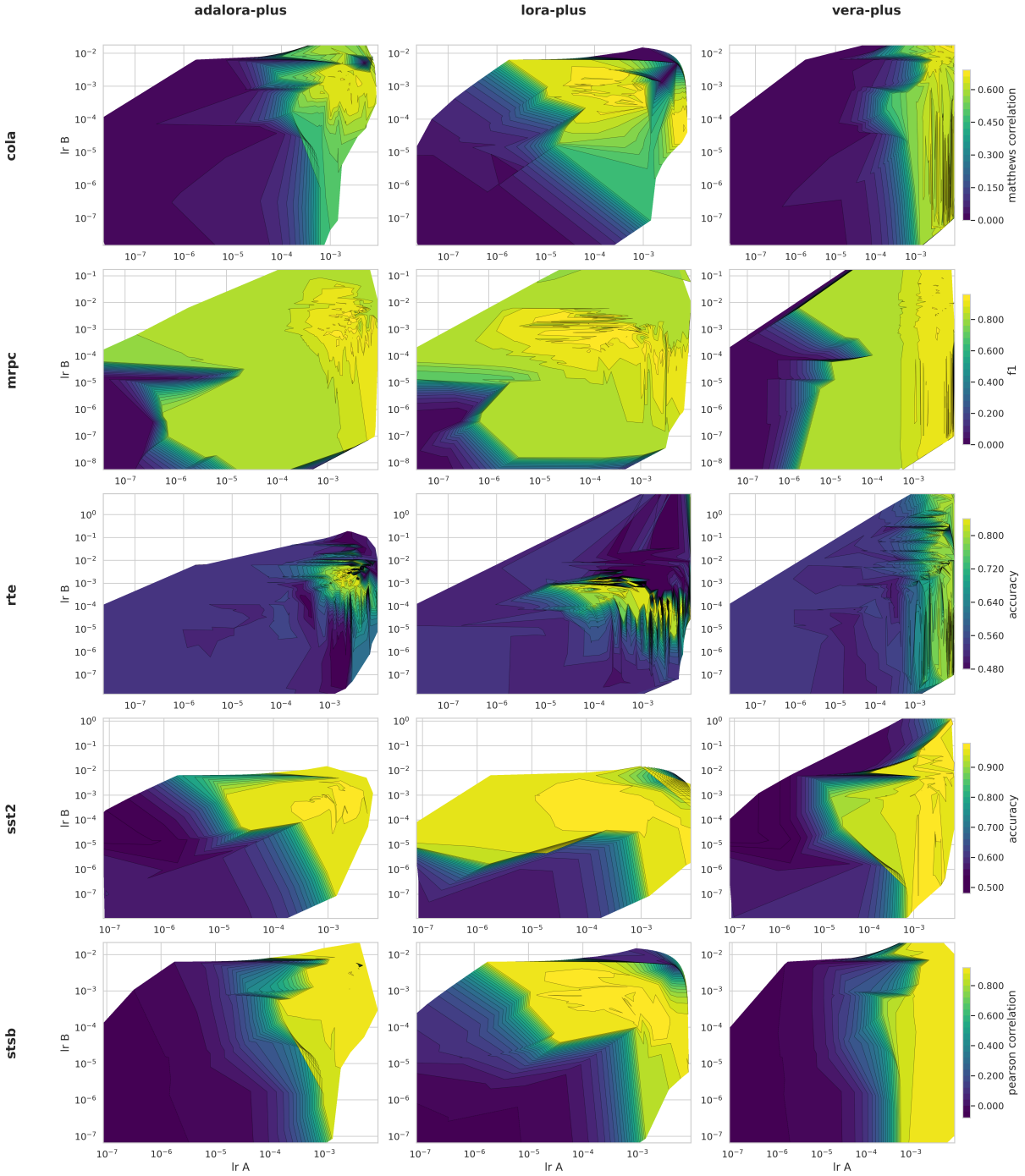

Figure 2: Search space visualization for RoBERTa-large across different adapters and tasks, constructed from all completed trials across all search spaces.

Table 3: Learning rate search results for CoLA, MRPC, and RTE across different models and adapters.

Model	Adapter	ρ Range	CoLA		MRPC		RTE	
			lr_a	lr_b	lr_a	lr_b	lr_a	lr_b
RoBERTa base	AdaLoRA	full	6.43E-03	1.77E-03	3.81E-03	2.30E-03	4.63E-03	2.59E-03
		$\rho \geq 1$	4.85E-03	5.26E-03	3.76E-03	3.78E-03	4.14E-03	4.24E-03
	AdaLoRA+	$\rho \leq 1$	5.27E-03	7.79E-04	3.04E-03	5.60E-04	4.52E-03	1.11E-03
		$\rho = 1$	2.48E-03	2.48E-03	3.22E-03	3.22E-03	3.32E-03	3.32E-03
	LoRA	full	6.26E-04	1.40E-03	1.59E-03	3.76E-04	3.74E-03	1.14E-04
		$\rho \geq 1$	3.59E-04	2.08E-03	3.05E-04	1.23E-03	7.20E-04	1.21E-03
	LoRA+	$\rho \leq 1$	6.42E-03	3.50E-04	4.23E-03	5.28E-04	1.14E-03	5.27E-04
		$\rho = 1$	8.23E-04	8.23E-04	1.52E-03	1.52E-03	1.13E-03	1.13E-03
	VeRA	full	9.97E-03	2.70E-07	9.41E-03	3.51E-07	6.56E-03	7.33E-08
		$\rho \geq 1$	4.98E-03	1.52E-02	4.65E-03	6.24E-03	4.99E-03	6.92E-03
	VeRA+	$\rho \leq 1$	3.70E-03	7.58E-04	8.03E-03	1.08E-06	8.52E-03	1.34E-04
		$\rho = 1$	9.69E-03	9.69E-03	7.81E-03	7.81E-03	5.94E-03	5.94E-03
RoBERTa large	AdaLoRA	full	3.62E-03	1.29E-03	7.89E-03	3.03E-03	2.90E-03	1.17E-03
		$\rho \geq 1$	1.57E-03	2.91E-03	1.33E-03	3.06E-03	2.11E-03	2.81E-03
	AdaLoRA+	$\rho \leq 1$	2.45E-03	1.14E-03	1.16E-03	6.47E-04	2.63E-03	1.15E-03
		$\rho = 1$	1.77E-03	1.77E-03	5.06E-03	5.06E-03	2.08E-03	2.08E-03
	LoRA	full	9.07E-05	1.97E-03	6.08E-04	4.03E-04	9.86E-03	2.98E-05
		$\rho \geq 1$	5.91E-05	1.14E-03	2.97E-04	9.15E-04	1.89E-04	1.15E-03
	LoRA+	$\rho \leq 1$	6.92E-03	5.47E-05	1.91E-03	8.11E-05	3.16E-03	1.39E-04
		$\rho = 1$	2.57E-04	2.57E-04	5.46E-04	5.46E-04	5.96E-04	5.96E-04
	VeRA	full	2.29E-03	3.18E-05	9.98E-03	4.07E-07	5.70E-03	4.30E+01
		$\rho \geq 1$	4.22E-03	1.76E-02	3.40E-03	4.45E-02	5.00E-03	6.93E-03
	VeRA+	$\rho \leq 1$	5.11E-03	1.72E-07	7.44E-03	1.04E-07	9.98E-03	3.72E-07
		$\rho = 1$	9.93E-03	9.93E-03	3.57E-03	3.57E-03	5.06E-03	5.06E-03

Table 4: Learning rate search results for SST-2 and STS-B across different models and adapters.

Model	Adapter	ρ Range	SST-2		STS-B	
			lr_a	lr_b	lr_a	lr_b
RoBERTa base	AdaLoRA	full	1.12E-03	4.56E-04	5.63E-03	1.83E-03
		$\rho \geq 1$	1.08E-03	1.49E-03	2.51E-03	3.93E-03
	AdaLoRA+	$\rho \leq 1$	2.37E-03	1.34E-03	3.49E-03	1.63E-03
		$\rho = 1$	1.57E-03	1.57E-03	3.50E-03	3.50E-03
	LoRA	full	5.25E-04	6.33E-04	1.47E-03	2.66E-04
		$\rho \geq 1$	2.26E-04	2.44E-03	2.58E-04	2.52E-03
	LoRA+	$\rho \leq 1$	1.41E-03	6.32E-04	1.49E-03	6.94E-04
		$\rho = 1$	1.02E-03	1.02E-03	4.28E-04	4.28E-04
	VeRA	full	7.87E-03	9.79E-05	5.75E-03	1.91E-01
		$\rho \geq 1$	4.62E-03	2.81E-02	4.94E-03	5.52E-03
	VeRA+	$\rho \leq 1$	5.40E-03	3.11E-05	8.87E-03	9.37E-06
		$\rho = 1$	7.53E-03	7.53E-03	7.27E-03	7.27E-03
RoBERTa large	AdaLoRA	full	9.82E-04	4.13E-04	1.96E-03	2.64E-03
		$\rho \geq 1$	8.53E-04	9.01E-04	2.29E-03	2.29E-03
	AdaLoRA+	$\rho \leq 1$	3.18E-03	3.57E-04	2.93E-03	1.12E-03
		$\rho = 1$	2.30E-03	2.30E-03	3.04E-03	3.04E-03
	LoRA	full	1.45E-04	2.84E-03	2.47E-04	6.02E-04
		$\rho \geq 1$	1.83E-04	9.07E-04	2.04E-04	9.00E-04
	LoRA+	$\rho \leq 1$	1.33E-03	2.20E-04	1.87E-03	9.31E-05
		$\rho = 1$	3.65E-04	3.65E-04	5.16E-04	5.16E-04
	VeRA	full	1.87E-03	3.02E-06	9.98E-03	1.41E-03
		$\rho \geq 1$	3.29E-03	2.73E-02	4.97E-03	1.02E-02
	VeRA+	$\rho \leq 1$	2.65E-03	5.48E-04	9.30E-03	1.95E-07
		$\rho = 1$	1.37E-03	1.37E-03	5.26E-03	5.26E-03