

PEFT Gains Need Hidden Baselines: Rank and Optimizer Sweeps Change LoRA Conclusions

Ding Wu

Independent Researcher

dingwu@alumni.upenn.edu

Abstract

New parameter-efficient fine-tuning (PEFT) methods are often credited with gains against a single default LoRA configuration, even though inexpensive tuning of vanilla LoRA may change the comparison. We present a deliberately small, descriptive case study of two claims: adaptive rank allocation with AdaLoRA and LoRA-specific optimization with LoRA-RITE. Using RoBERTa-base on capped subsets of three GLUE classification tasks, we compare each method with a pre-specified vanilla LoRA control sweep. AdaLoRA beats fixed LoRA at rank 8 in 4 of 7 dataset–seed cells, but beats the best tested fixed rank in only 1 of 7. LoRA-RITE beats default AdamW in 3 of 6 cells, but tuned AdamW in only 1 of 6. Several SST-2 runs collapse to the majority label, so these counts are illustrative rather than population-level estimates. The practical contribution is an attribution protocol: report modest vanilla rank/learning-rate sweeps, per-cell margins, instability checks, and budget accounting before attributing gains to a specialized PEFT mechanism.

1 Introduction

Parameter-efficient fine-tuning adapts a pretrained model by training a small set of added or selected parameters while leaving most of the base model frozen (Houlsby et al., 2019; Hu et al., 2022). LoRA is especially common because it represents task updates with low-rank matrices and adds no inference-time module after merging (Hu et al., 2022). Recent methods improve this recipe through adaptive rank allocation, such as AdaLoRA (Zhang et al., 2023), or through optimizers specialized for LoRA’s factorization, such as LoRA-RITE (Yen et al., 2025).

These comparisons depend on what counts as the vanilla LoRA baseline. A method can look useful against one default rank and learning rate but less distinctive once vanilla LoRA receives a small,

method-agnostic sweep. We call this omitted control a *hidden baseline*: a pre-specified sweep that is cheap enough to report but is often absent from the headline comparison. It is neither a new method nor an oracle over an unlimited search space.

We audit this issue in a low-budget encoder-classification setting. GLUE is a standard collection of natural-language understanding benchmarks (Wang et al., 2019); we use SST-2 sentiment classification, MRPC paraphrase classification, and RTE textual entailment. The setting is intentionally compressed—512 training examples and two epochs—so it should be read as a stress test of attribution, not as a definitive ranking of PEFT methods.

The central contribution is an evaluation insight: under modest vanilla controls, the apparent source of improvement changes. Rank and learning-rate sweeps substantially alter the conclusions drawn from default-only comparisons, while raw prediction checks expose instability that averages would hide. We therefore propose a lightweight reporting protocol that makes baseline tuning and failures visible without requiring an unlimited hyperparameter search.

2 Related Work

PEFT and LoRA. Adapters introduced a compact alternative to full fine-tuning (Houlsby et al., 2019). LoRA later represented updates with low-rank factors while freezing pretrained weights (Hu et al., 2022).

Rank allocation. AdaLoRA allocates a rank budget using parameter importance and prunes singular directions during training (Zhang et al., 2023). Related work studies flexible, salience-based, and gradient-driven rank allocation (Ke et al., 2024; He et al., 2025; Liu et al., 2026). If rank choice is central enough to motivate a new mechanism, a small fixed-rank sweep is a natural attribution control.

LoRA optimization. LoRA-RITE proposes a transformation-invariant optimizer for low-rank factors (Yen et al., 2025). LoRA-Pro and ALLoRA likewise show that optimization and learning-rate behavior are active design questions (Wang et al., 2025; Huang and Balestrierio, 2024). Our question is complementary: does a specialized optimizer remain advantageous when vanilla LoRA with AdamW receives a small rank/LR sweep?

3 Experimental Setup

We use RoBERTa-base (Liu et al., 2019) with LoRA-style PEFT. Metrics follow common GLUE practice: accuracy for SST-2 and RTE, and F1 for MRPC. Every run uses 512 training examples, 500 validation examples, 2 epochs, batch size 16, and the same base-model cache.

Dynamic-rank audit. We compare fixed LoRA ranks $r \in \{4, 8, 16, 32\}$ with AdaLoRA target ranks $\{8, 16\}$. The default fixed baseline is $r = 8$. The seven dataset-seed cells are SST-2 seeds 42/43, MRPC seeds 42/43, and RTE seeds 42/43/44.

Optimizer audit. We compare AdamW LoRA and LoRA-RITE at ranks 8 and 16. AdamW uses learning rates $\{10^{-4}, 2 \times 10^{-4}, 5 \times 10^{-4}\}$; LoRA-RITE uses $\{10^{-4}, 2 \times 10^{-4}\}$. The default AdamW baseline is $r = 8$, learning rate 2×10^{-4} . The six cells are SST-2, MRPC, and RTE crossed with seeds 42/43.

Why the search spaces are asymmetric. Fixed LoRA receives four ranks while AdaLoRA receives two target ranks; AdamW receives six rank/LR configurations while LoRA-RITE receives four. We did not add post-hoc trials to equalize grid density because the original study was designed to ask a narrower audit question: can a modest vanilla control change attribution? The tuned baseline is therefore a diagnostic control, not a compute-matched tournament. Equal-density or equal-compute tuning remains necessary before ranking the methods themselves.

4 Results

4.1 Default baselines look more favorable

AdaLoRA beats fixed LoRA at the default rank in 4/7 cells, but the best tested fixed rank in only 1/7. LoRA-RITE beats default AdamW in 3/6 cells, but tuned AdamW in only 1/6. These counts describe the tested cells; they are not precise population win

rates. Wilson 95% intervals are correspondingly wide: 1/7 gives 0.03–0.51 and 1/6 gives 0.03–0.56.

The mean AdamW rank/LR sweep gain over the default is +0.1055, whereas the mean LoRA-RITE effect is -0.0013 against default AdamW and -0.1068 against tuned AdamW. For AdaLoRA, the mean effect is -0.0963 against fixed $r = 8$ and -0.1102 against the best tested fixed rank.

4.2 Rank sweeps shrink adaptive-rank gains

AdaLoRA sometimes improves over $r = 8$, especially outside SST-2. However, the best fixed rank usually matches or exceeds the best AdaLoRA target. The surviving nonnegative margins against the tuned rank are small outside the collapsed SST-2 cells (Table 2). This does not show that adaptive rank is useless; it shows that a single fixed rank is insufficient evidence for attributing gains to adaptive allocation.

4.3 Learning-rate sensitivity and compressed data

The optimizer result is not a cosmetic consequence of selecting among nearly identical settings. On SST-2, AdamW at 5×10^{-4} reaches 0.8920/0.8780 accuracy, whereas the default AdamW cells are 0.6980/0.5360. The compressed regime can amplify optimization sensitivity: with only 512 training examples and two epochs, some configurations fail to leave the majority-class solution. We therefore interpret the result as evidence that default-only attribution is fragile under this budget, not as an explanation of sensitivity in larger-scale LLM tuning.

4.4 Sanity check: label collapse, not a parser bug

We reran eight suspicious SST-2 cells with raw prediction export. Saved metrics match raw prediction accuracy in every sanity run. Four AdaLoRA runs and one LoRA-RITE run predict label 1 for all 500 validation examples, matching the 265/500 majority-class baseline and producing 0.5300 accuracy. The tuned AdamW controls do not collapse and reach 0.8920/0.8780. These are observed failures under the tested budget, not universal properties of the methods.

5 Recommendations

A minimal PEFT comparison should: (1) declare a small vanilla rank/LR sweep before examining outcomes; (2) report default and tuned controls; (3)

Audit	Method	Search grid	Learning rate / optimizer	Runs
Rank	Fixed LoRA	$r = \{4, 8, 16, 32\}$	8×10^{-4} / AdamW	28
Rank	AdaLoRA	target $\{8, 16\}$	8×10^{-4} / AdamW	14
Optimizer	AdamW LoRA	$r = \{8, 16\}$, LR $\{1, 2, 5\} \times 10^{-4}$	AdamW	36
Optimizer	LoRA-RITE	$r = \{8, 16\}$, LR $\{1, 2\} \times 10^{-4}$	LoRA-RITE	24

Table 1: Search budget. The baseline sweep is an attribution control; the grids are intentionally small and not symmetric.

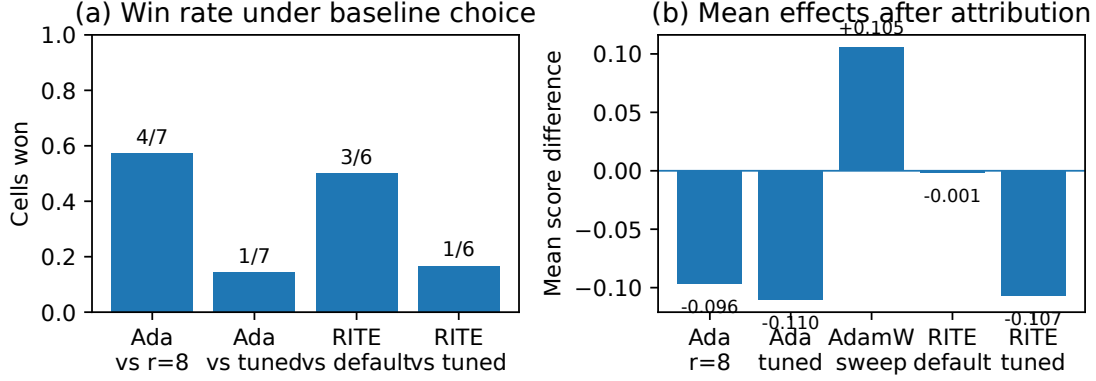


Figure 1: Headline audit. Specialized variants look more favorable against default vanilla settings than against modestly tuned controls. The AdamW sweep gain is itself larger than the mean LoRA-RITE effect against default AdamW.

Cell	AdaLoRA–best fixed	RITE–best AdamW
MRPC-42	+0.0050	+0.0034
MRPC-43	-0.0001	-0.0005
RTE-42	-0.0036	-0.0144
RTE-43	-0.0144	-0.0614
RTE-44	+0.0000	–
SST-2-42	-0.3820	-0.3620
SST-2-43	-0.3760	-0.2060

Table 2: Per-cell margins against tuned hidden baselines. Positive values mean the specialized method wins.

show per-cell margins and seed variation, rather than only averages; (4) disclose whether search is equalized by run count or compute; and (5) preserve unstable cells with prediction distributions instead of silently filtering them. The protocol is not a mandate for unlimited tuning. Its purpose is to prevent a cheap missing control from being mistaken for a method-specific gain.

6 Conclusion

A small, method-agnostic sweep over vanilla LoRA can materially change the attribution of apparent PEFT gains in this low-budget setting. The result is not a ranking of AdaLoRA or LoRA-RITE; it is evidence that default-only baselines are too fragile to support strong mechanism claims without modest tuning and failure diagnostics.

Limitations

This study uses capped GLUE subsets, RoBERTa-base, encoder classification, limited seeds, and only three task families. GLUE is mature rather than current, but its small, standardized tasks make failure auditing and raw-prediction verification tractable; results should not be extrapolated to modern instruction tuning without replication. The study is not a full reproduction of AdaLoRA or LoRA-RITE at their original scale. Search spaces are small and asymmetric, and the number of cells is too small for strong statistical claims. Some gaps may reflect robustness under compressed training rather than intrinsic method quality. Peak GPU memory was not logged for every run. Most importantly, the result supports a reporting recommendation, not a broad dismissal of specialized PEFT methods.

Acknowledgments

We thank the Insights 2026 reviewers for suggestions that improved the framing of the small- n evidence, clarified the asymmetric search budget, and motivated a fuller discussion of learning-rate sensitivity and the compressed training regime.

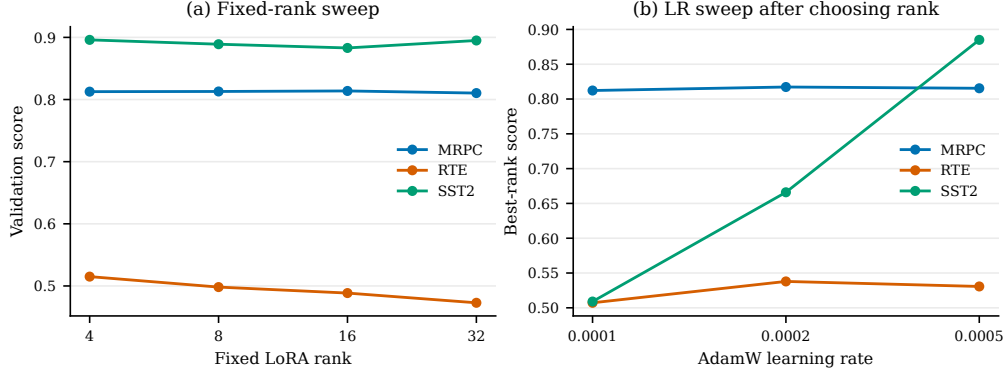


Figure 2: Hidden baselines come from small, method-agnostic sweeps. Left: changing fixed LoRA rank changes the baseline. Right: AdamW learning-rate tuning, after selecting the best tested rank at each learning rate, changes the optimizer comparison.

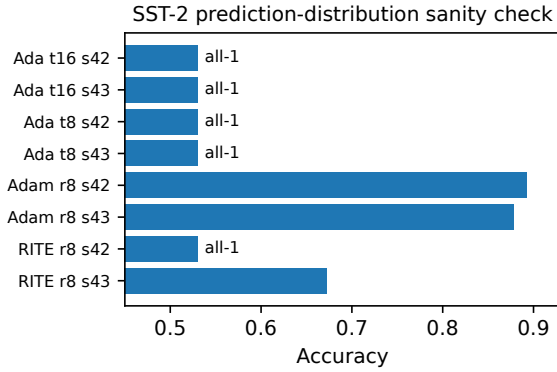


Figure 3: SST-2 sanity audit. Runs marked “all-1” predict the positive label for all 500 validation examples.

References

- Haonan He, Peng Ye, Yuchen Ren, Yuan Yuan, Luyang Zhou, Shucun Ju, and Lei Chen. 2025. GoRA: Gradient-driven adaptive low rank adaptation. *arXiv:2502.12171*.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of ICML*, pages 2790–2799.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *ICLR*.
- Hai Huang and Randall Balestriero. 2024. ALLoRA: Adaptive learning rate mitigates LoRA fatal flaws. *arXiv:2410.09692*.
- Wenjun Ke, Jiahao Wang, Peng Wang, Jiajun Liu, Dong Nie, Guozheng Li, and Yining Li. 2024. Unveiling LoRA intrinsic ranks via salience analysis. In *NeurIPS*.
- Muqing Liu, Chongjie Si, and Yuheng Jia. 2026. FlexLoRA: Entropy-guided flexible low-rank adaptation. In *ICLR*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv:1907.11692*.
- 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. In *ICLR*.
- Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. 2025. LoRA-Pro: Are low-rank adapters properly optimized? In *ICLR*.
- Jui-Nan Yen, Si Si, Zhao Meng, Felix Yu, Sai Surya Duvvuri, Inderjit S. Dhillon, Cho-Jui Hsieh, and Sanjiv Kumar. 2025. LoRA done RITE: Robust invariant transformation equilibration for LoRA optimization. In *ICLR*.
- 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. In *ICLR*.

A Full Dynamic-Rank Cell Table

Task	Seed	$r = 8$	Best fixed	Best r	Best AdaLoRA	Ada- $r = 8$	Ada-best
MRPC	42	.8173	.8173	8	.8223	+0.0050	+0.0050
MRPC	43	.8086	.8154	16	.8152	+0.0067	-0.0001
RTE	42	.5199	.5307	4	.5271	+0.0072	-.0036
RTE	43	.5018	.5415	4	.5271	+0.0253	-.0144
RTE	44	.4729	.4729	4	.4729	+0.0000	+0.0000
SST-2	42	.8920	.9120	4	.5300	-.3620	-.3820
SST-2	43	.8860	.9060	32	.5300	-.3560	-.3760

Table 3: Full dynamic-rank comparison. MRPC uses F1; SST-2 and RTE use accuracy.

B Full LoRA-RITE Cell Table

Task	Seed	Default AdamW	Tuned AdamW	Tuned config	Best RITE	RITE-default	RITE-tuned
MRPC	42	.8122	.8223	$r = 16, LR=2e-4$	.8257	+0.0135	+0.0034
MRPC	43	.8122	.8127	$r = 8, LR=5e-4$	.8122	+0.0000	-.0005
RTE	42	.5415	.5415	$r = 8, LR=2e-4$	.5271	-.0144	-.0144
RTE	43	.5018	.5884	$r = 8, LR=5e-4$	.5271	+0.0253	-.0614
SST-2	42	.6980	.8920	$r = 8, LR=5e-4$	.5300	-.1680	-.3620
SST-2	43	.5360	.8780	$r = 8, LR=5e-4$	.6720	+0.1360	-.2060

Table 4: Full LoRA-RITE comparison. MRPC uses F1; SST-2 and RTE use accuracy.

C SST-2 Sanity Details

Run	Acc.	Pred 0	Pred 1	Gold 0	Gold 1	Assessment
AdaLoRA target 8 seed 42	.5300	0	500	235	265	label collapse
AdaLoRA target 8 seed 43	.5300	0	500	235	265	label collapse
AdaLoRA target 16 seed 42	.5300	0	500	235	265	label collapse
AdaLoRA target 16 seed 43	.5300	0	500	235	265	label collapse
AdamW tuned seed 42	.8920	213	287	235	265	metric pipeline sane
AdamW tuned seed 43	.8780	244	256	235	265	metric pipeline sane
LoRA-RITE seed 42	.5300	0	500	235	265	label collapse
LoRA-RITE seed 43	.6720	85	415	235	265	not full collapse

Table 5: SST-2 prediction-distribution sanity check. The saved metric matches raw prediction accuracy in every retained row.

D Budget Accounting

The budget accounting behind Table 1 records audit family, task, seed, method, rank or target rank, optimizer, learning rate, trainable parameters, training time, and peak-memory status. Peak GPU memory is reported only when present in the recovered source results; missing memory values are not imputed. This preserves the distinction between measured resource use and unavailable logging.

E Additional Figures

The following diagnostics reproduce the auxiliary views used in the accepted version. They are descriptive checks rather than additional experiments.

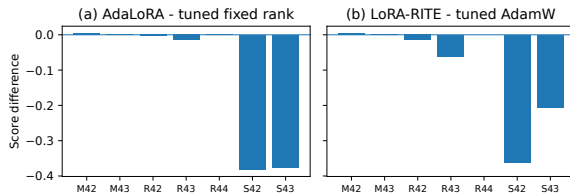


Figure 4: Cell-level score differences against tuned hidden baselines. Positive values mean the specialized method wins the dataset-seed cell after the vanilla baseline sweep.

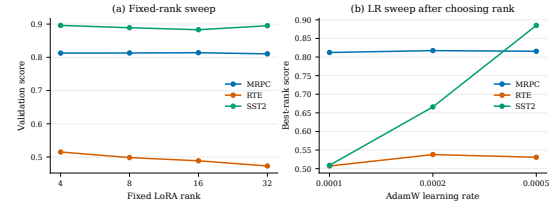


Figure 5: Rank and learning-rate sweeps used to define the tuned hidden baselines.

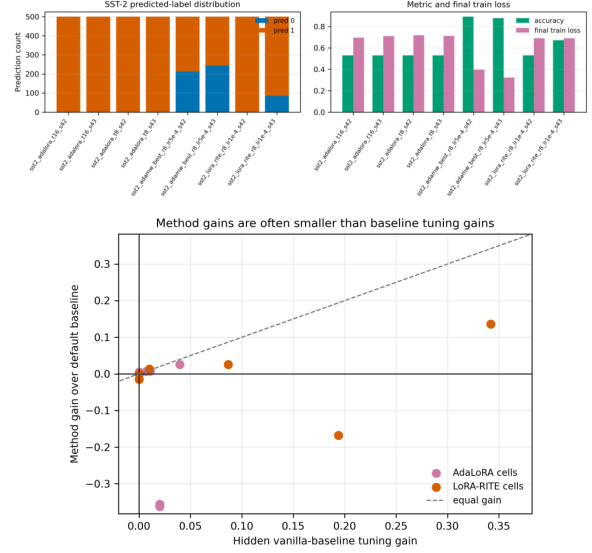


Figure 6: SST-2 failure diagnostics and the relation between method gain and hidden-baseline tuning gain.

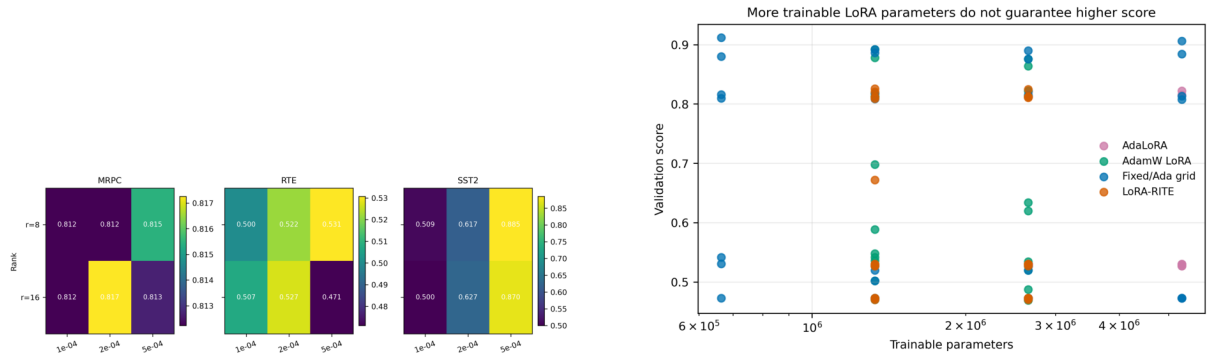


Figure 7: Additional diagnostic views for the optimizer grid and parameter budget. Learning-rate sensitivity is visible, while trainable-parameter count alone does not determine validation score.

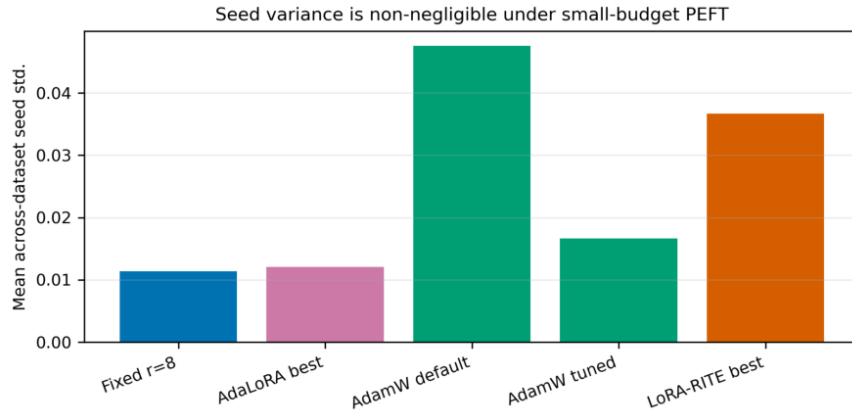


Figure 8: Seed variance is non-negligible under the small-budget PEFT setting, motivating cell-level and seed-aware reporting alongside means.