Gradient-Based LoRA Rank Allocation Under GRPO: An Empirical Study

Anonymous Authors[1]

Abstract

Adaptive rank allocation for LoRA — allocating more parameters to important layers and fewer to unimportant ones — consistently improves efficiency under supervised fine-tuning (SFT). We investigate whether this success transfers to reinforcement learning, specifically Group Relative Policy Optimization (GRPO). Using gradient-magnitude profiling on Qwen 2.5 1.5B with GSM8K, we find that it does not: proportional rank allocation *degrades* accuracy by 4.5 points compared to uniform allocation (70.0% vs. 74.5%), despite using identical parameter budgets. We identify two mechanisms behind this failure. First, the gradient landscape under GRPO is fundamentally flatter than under SFT — the max-to-min layer importance ratio is only 2.17×, compared to >10× reported in SFT literature. All layers carry meaningful gradient signal; none are truly idle. Second, we discover a *gradient amplification effect*: non-uniform allocation widens the importance spread from 2.17× to 3.00×, creating a positive feedback loop where high-rank layers absorb more gradient while low-rank layers are progressively silenced. Our results suggest that gradient importance does not predict capacity requirements under RL, and that naïve transfer of SFT-era rank allocation to alignment training should be avoided.

1. Introduction

Parameter-efficient fine-tuning via Low-Rank Adaptation (Hu et al., 2022) has become the standard approach for adapting large language models. LoRA decomposes weight updates into low-rank matrices $\Delta W = BA$ with uniform rank r across all layers.

Recent work has challenged this uniform assumption. AdaLoRA (Zhang et al., 2023) dynamically prunes singular values during training based on importance scores. GoRA (He et al., 2025) allocates rank proportionally to gradient-weight products at initialization. Aletheia (Saket, 2026) selects layers via lightweight gradient probes. ILA (Shi et al., 2024) shows that only 10–30% of layers are significant for alignment under SFT. These methods achieve meaningful efficiency gains, establishing that **gradient importance correlates with capacity requirements under supervised objectives**.

A natural question follows: does this correlation hold under reinforcement learning? RL-based alignment methods like GRPO (Shao et al., 2024) optimize a fundamentally different objective — advantage-weighted policy gradients with sparse, binary reward signals rather than dense per-token cross-entropy loss. Theoretical work suggests that RL gradients concentrate differently than SFT gradients (Young, 2026), motivating the hypothesis that rank allocation strategies should differ.

We test this hypothesis by applying gradient-based rank profiling to GRPO training and find a surprising result: **adaptive rank allocation hurts performance under RL**. Our investigation reveals three findings:

1. **Flat gradient landscape:** Under GRPO, layer importance is distributed far more uniformly than under SFT (2.17× max/min ratio vs. >10×). All layers are load-bearing.

2. **Gradient amplification:** Non-uniform allocation creates a positive feedback loop — high-rank layers absorb more gradient while low-rank layers are silenced, widening the spread from 2.17× to 3.00×.

3. **Generalization gap:** Models train equally well (identical reward curves) but generalize differently — the damage from rank reallocation appears only at evaluation time.

[1]**AUTHORERR: Missing \icmlaffiliation. .AUTHORERR: Missing \icmlcorrespondingauthor.**

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Method

2.1. GRPO with LoRA

GRPO generates K completions per prompt, computes rewards, and normalizes advantages within the group:

$$\hat{A}_i = \frac{r_i - \mu_{\text{group}}}{\sigma_{\text{group}}} \tag{1}$$

When combined with LoRA, gradients flow through adapter parameters $B^{(l)}, A^{(l)}$ at each layer l.

2.2. Reward Sensitivity Profiling

We define the reward sensitivity score for layer l as the mean gradient norm over T training steps:

$$S(l) = \frac{1}{T} \sum_{t=1}^{T} \sum_{m \in \mathcal{M}} \left\| \nabla_{\theta_m^{(l)}} \mathcal{L}_t \right\|_2 \tag{2}$$

where $\mathcal{M}$ is the set of target modules (q/k/v/o/up/down/gate projections). This score captures how much each layer's parameters respond to the GRPO reward signal.

2.3. Rank Allocation

Given total rank budget $R_{\text{total}} = L \times r_{\text{uniform}}$, we allocate per-layer rank:

$$r^{(l)} = \text{clip}\left(\text{round}\left(\frac{S(l)}{\sum_{l'} S(l')} \times R_{\text{total}}\right), r_{\min}, r_{\max}\right) \tag{3}$$

with $r_{\min} = 4$ and $r_{\max} = 64$, rounded to multiples of 4. We also evaluate random allocation as a control.

3. Experiments

3.1. Setup

Model: Qwen/Qwen2.5-1.5B-Instruct (28 transformer layers). **Dataset:** GSM8K (Cobbe et al., 2021) with structured XML output format. **Rewards:** Format compliance (1.0 for correct `<think>/<answer>` tags) and answer correctness (1.0 for correct numerical answer). **LoRA:** Applied to all 7 projection modules per layer (196 adapters total). Uniform baseline: $r = 32$. **GRPO:** $K = 4$ generations, $\beta = 0.05$ KL coefficient, 1000 training steps, learning rate 10^{-5}, vLLM colocate generation.

3.2. Gradient Landscape Under GRPO

Figure 1 shows the reward sensitivity map. Key observations:

- **Flat distribution:** Max/min importance ratio is 2.17× (Layer 15 hottest at 4.68%, Layer 26 coldest at 2.15%). Even the coldest layer carries 46% of the hottest layer's gradient signal.
- **Middle-layer concentration:** Layers 9–18 carry 43.0% of gradient, but early (29.8%) and late (27.2%) layers remain meaningful — unlike SFT where ILA (Shi et al., 2024) reports >80% concentration in the top 30%.
- **Temporal stability:** Early-vs-late training correlation is 0.962, indicating stable structural patterns, not transient noise.
- **Module importance:** Attention (52.9%) and FFN (47.1%) contribute roughly equally. The `up_proj` module is most reward-sensitive (21.4%).

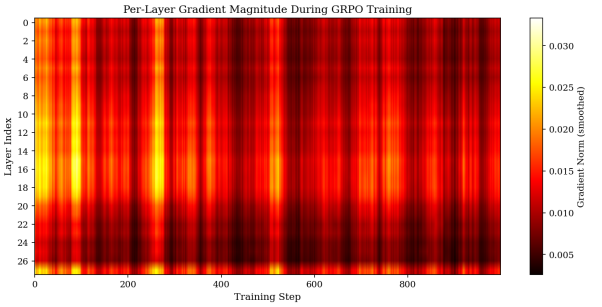

Figure 1. Per-layer gradient magnitude during GRPO training (1000 steps, 28 layers). The distribution is notably flat compared to SFT-era findings.

Table 1. GSM8K accuracy under different rank allocations (n=200 test samples). All "same budget" methods use total rank 896 (28 × 32). Confidence intervals are Wilson score intervals.

Strategy	Params	Acc. (%)	95% CI
Base model (no LoRA)	0	66.0	±6.6
Uniform (r=32)	36.9M	**74.5**	±6.0
Proportional (r=20–40)	36.9M	70.0	±6.4
Random (r=16–48)	36.9M	67.5	±6.5
Reduced 70% (r=12–31)	25.8M	65.0	±6.6

3.3. Rank Allocation Results

Table 1 shows our main result: **uniform allocation outperforms all non-uniform variants**, including gradient-aware proportional allocation. Proportional allocation with identical parameter budget scores 4.5 points below uniform. Random allocation — a control with non-uniform ranks but no gradient guidance — scores even lower at 67.5%. Reduced-budget allocation performs below the untrained base model.

Two observations stand out. First, **gradient-aware allocation outperforms random** (70.0% vs. 67.5%), confirming that the importance signal is meaningful — it identifies

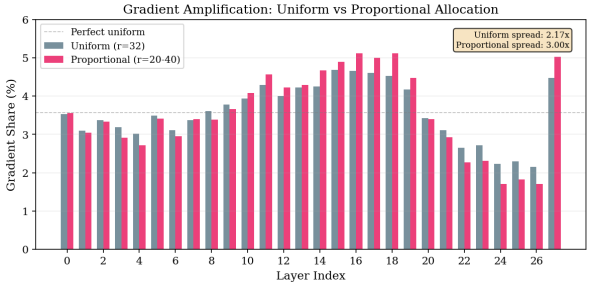

Figure 2. Normalized layer importance under uniform vs. proportional allocation. Non-uniform allocation amplifies the original importance spread from 2.17× to 3.00×.

genuinely important layers. However, both lose to uniform, suggesting that while the signal is directionally correct, any deviation from uniform allocation damages performance under GRPO.

Second, training reward curves are nearly identical across all configurations — correctness reward reaches 0.74–0.77 for all methods by step 1000. The performance gap appears *only at evaluation time*, indicating that rank reallocation damages generalization rather than training dynamics. Models learn equally well on the training distribution but differ in their ability to transfer to unseen problems.

3.4. The Gradient Amplification Effect

We profile gradients during *all* training runs, not just the uniform baseline. Figure 2 reveals a striking effect: the gradient importance spread *widens* under non-uniform allocation.

- Uniform: max/min ratio 2.17×
- Proportional (same budget): 3.00× (+38%)
- Reduced 70%: 3.57× (+64%)

Layers given higher rank (e.g., Layer 18: r=40) see their gradient share *increase* from 4.53% to 5.12%. Conversely, layers given lower rank (e.g., Layer 24: r=20) see their share *decrease* from 2.23% to 1.71%. The allocation creates a positive feedback loop: more capacity → more gradient → appears even more "important."

Crucially, this effect is **causal, not correlative**. The random allocation experiment — where ranks bear no relation to gradient importance — shows equally strong amplification. The correlation between allocated rank and resulting gradient shift is 0.972 for random allocation and 0.946 for proportional. Under random allocation, Layer 1 (normally 3.09% of gradient) receives rank 48 and jumps to 4.21%; Layer 10 (normally 3.93%) receives rank 16 and drops to 2.85%. **Rank determines gradient importance, not the other way around.**

This means gradient profiling under one allocation cannot reliably inform a different allocation — the "profile then retrain" paradigm is fundamentally flawed for GRPO. One might suspect this is trivially explained by more parameters yielding larger aggregate gradient norms. However, our reward sensitivity score (Eq. 2) normalizes by module count, and the amplification persists even when examining individual modules at identical dimensions (e.g., `q_proj` at 1536×1536 across all layers). The effect reflects genuine changes in how the training signal distributes across the network, not merely a parameter-counting artifact.

This amplification effect has not been reported in prior work, likely because SFT-based methods show the opposite pattern — AdaLoRA's dynamic pruning stabilizes importance distributions by adjusting continuously during training rather than committing to a fixed allocation.[1]

3.5. Why SFT Methods Fail Under RL

The fundamental difference lies in gradient distribution:

Under SFT, the loss is per-token cross-entropy where a small number of layers dominate the gradient landscape. ILA (Shi et al., 2024) shows the top 30% of layers carry >80% of gradient signal, and freezing the rest improves performance. This concentrated structure makes adaptive allocation effective — there are genuinely idle layers whose capacity can be safely redistributed.

Under GRPO, the loss is advantage-weighted policy gradient with sparse, binary reward. Our profiling reveals a fundamentally flatter landscape where the top 30% carry only 35.7% of signal. Even the "coldest" layers contribute meaningfully (2.15% of total, or 46% of the hottest). Reducing their capacity — even modestly, from r=32 to r=20 — damages evaluation accuracy while leaving training reward unchanged.

We hypothesize that low-gradient layers handle essential structural functions under RL: output formatting, numerical precision, and coherence maintenance. These functions generate small gradients because they are already well-handled by the pretrained model, but they become bottlenecks when capacity is reduced.

This hypothesis is supported by our module-level analysis: attention and FFN modules contribute 52.9% and 47.1% respectively, a near-even split unlike SFT where FFN layers in the top half of the network dominate (Zhang et al., 2023).

[1]We also discovered that PEFT's `rank_pattern` wildcard matching (e.g., `model.layers.*.q_proj`) silently fails for Qwen models, falling back to default rank. Exact module paths (e.g., `model.layers.N.self_attn.q_proj`) are required.

The up_proj module is most reward-sensitive (21.4%), while q_proj contributes only 9.5% — yet both are essential for correct mathematical reasoning.

Our findings also reveal that the gradient importance map under GRPO is temporally stable (early-vs-late training correlation: 0.962), ruling out the possibility that a different profiling window would yield a more useful allocation. The flat landscape is a structural property of how GRPO distributes learning, not an artifact of averaging over noisy training phases.

4. Related Work

Adaptive rank for SFT. AdaLoRA (Zhang et al., 2023) prunes singular values during training. GoRA (He et al., 2025) uses gradient-weight products for initialization-time allocation. IGU-LoRA (Cui et al., 2026) applies integrated gradients with uncertainty-aware scoring. Aletheia (Saket, 2026) selects layers via gradient probes. **All operate exclusively under supervised objectives.**

Layer importance in alignment. ILA (Shi et al., 2024) learns binary layer masks showing 10–30% of layers suffice for SFT alignment. Young (2026) prove that RLHF gradients concentrate at specific positions. Our work extends this line by showing that RL's gradient concentration is *insufficient* for effective rank allocation.

GRPO and LoRA. DeepSeekMath (Shao et al., 2024) introduced GRPO for mathematical reasoning. To our knowledge, no prior work has investigated adaptive rank allocation specifically under RL-based alignment methods.

5. Limitations

Our findings are derived from a single model (Qwen 2.5 1.5B), a single dataset (GSM8K), and a single RL algorithm (GRPO). Whether these results generalize to larger models, other domains (code, safety alignment), or other RL methods (PPO, DPO) remains an open question. Our evaluation uses n=200 test samples with single-seed runs; the confidence intervals in Table 1 overlap, and we encourage replication at larger scale with multiple seeds to establish statistical significance. We compare to SFT gradient distributions by citing prior work rather than running a direct SFT baseline on the same model and data — a head-to-head comparison would strengthen the contrast. Finally, our hypothesis that cold layers handle structural functions (formatting, numerical precision) is untested; per-reward gradient decomposition could partially verify this.

6. Conclusion

We investigated whether gradient-based rank allocation — a proven technique for SFT — transfers to reinforcement learning alignment via GRPO. Our experiments show that it does not: proportional allocation degrades accuracy by 4.5 points despite identical parameter budgets. We identify the flat gradient landscape under GRPO and the gradient amplification effect as key mechanisms behind this failure.

These findings have practical implications: practitioners should not naïvely apply SFT-era rank allocation strategies to RL training. A random control confirms that gradient-aware allocation (70.0%) outperforms uninformed allocation (67.5%), validating that the importance signal is real — but insufficient to overcome the fundamental need for uniform capacity under GRPO.

Future work should explore *dynamic* rank adaptation methods that adjust continuously during training (avoiding the "profile then retrain" problem) and investigate whether the amplification effect — where rank causally determines gradient importance with $r > 0.97$ correlation — can be exploited for targeted capacity expansion rather than reallocation.

References

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Cui, X., Li, H., Zeng, R., Zhao, Y., Qian, J., Duan, W., Liu, B., and Zhou, Z. IGU-LoRA: Adaptive rank allocation via integrated gradients and uncertainty-aware scoring. *arXiv preprint arXiv:2603.13792*, 2026.

He, H., Ye, P., Ren, Y., Yuan, Y., Zhou, L., Ju, S., and Chen, L. GoRA: Gradient-driven adaptive low rank adaptation. In *Advances in Neural Information Processing Systems*, 2025.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Saket, A. Aletheia: Gradient-guided layer selection for efficient LoRA fine-tuning across architectures. *arXiv preprint arXiv:2604.15351*, 2026.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Shi, G., Lu, Z., Dong, X., Zhang, W., Zhang, X., Feng, Y., and Wu, X.-M. Understanding layer significance in LLM alignment. *arXiv preprint arXiv:2410.17875*, 2024.

Young, R. Why is RLHF alignment shallow? A gradient analysis. *arXiv preprint arXiv:2603.04851*, 2026.

Zhang, Q., Chen, M., Bukharin, A., He, P., Cheng, Y., Chen, W., and Zhao, T. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023.