Optimizing Financial Sentiment Analysis: A Systematic Study of LoRA Rank Selection

Raja Hussain[1,2*]

[1]New York University, New York, NY, USA

[2]ValtricAI Research, New York, NY, USA

research@valtric.ai

December 2025

Abstract

We present a systematic investigation of Low-Rank Adaptation (LoRA) rank selection for financial sentiment analysis, validated across multiple random seeds ($n = 5$ per configuration, 30 total runs). Using DistilRoBERTa-base (82M parameters) on the Twitter Financial News Sentiment dataset, we evaluate full fine-tuning and LoRA at ranks 4, 8, 16, 32, and 64. Our key finding is that **diminishing returns begin at rank 16**: while transitions from $r = 4 \to 8 \to 16$ show statistically significant improvements ($p < 0.01$), gains from $r = 16 \to 32 \to 64$ are **not statistically significant** ($p > 0.1$). Notably, the difference between $r = 32$ (84.4%) and $r = 64$ (84.6%) is indistinguishable from noise ($p = 0.35$). All LoRA configurations retain 94.6–97.4% of full fine-tuning accuracy while reducing trainable parameters by 97.8–99.2%. These results provide evidence-based guidance for practitioners: $r = 16$ **offers the best accuracy within the statistically significant improvement zone**, challenging the common practice of defaulting to higher ranks.

1 Introduction

1.1 Motivation

Financial sentiment analysis is critical for algorithmic trading, risk assessment, and market intelligence. However, deploying fine-tuned language models at scale presents challenges:

1. **Computational Cost**: Full fine-tuning of transformer models requires substantial GPU memory and training time.
2. **Multi-Model Deployment**: Financial institutions often need multiple specialized models (earnings, news, social media).
3. **Reproducibility**: Single-run evaluations can be misleading due to random variance.

Low-Rank Adaptation (LoRA) [Hu et al., 2022] has become the dominant parameter-efficient fine-tuning (PEFT) method, reducing trainable parameters by orders of magnitude while maintaining competitive performance. However, the relationship between LoRA rank and downstream task performance remains underexplored, particularly regarding *statistical significance* of observed differences.

[*]This work was conducted independently. It does not represent the views of New York University.

1.2 Research Question

What is the optimal LoRA rank for financial sentiment classification, and are differences between rank configurations statistically significant or merely noise?

1.3 Contributions

1. A **multi-seed validated** comparison ($n = 5$ per configuration) of LoRA ranks against full fine-tuning for financial sentiment analysis.
2. Evidence that $r = 32$ **and** $r = 64$ **are statistically indistinguishable** ($p = 0.35$), contradicting single-run conclusions.
3. Identification of $r = 16$ **as the inflection point** where statistically significant improvements cease.
4. Practical deployment recommendations grounded in statistical evidence rather than point estimates.

2 Related Work

2.1 Parameter-Efficient Fine-Tuning

Hu et al. [2022] introduced LoRA, arguing that weight updates during fine-tuning lie in a low-dimensional subspace. By injecting trainable low-rank decomposition matrices, LoRA reduces trainable parameters while preserving performance.

Zhao et al. [2024] evaluated LoRA at scale (310 models across 31 tasks), reporting large average gains over base models and strong practical deployment implications.

2.2 Distributed and Structured LoRA

Gao and Zhang [2024] proposed DLoRA for distributed PEFT. Their results motivate studying which LoRA capacity (rank) is actually useful for downstream tasks.

2.3 Financial Sentiment Analysis

FinBERT [Araci, 2019] established transformer-based approaches for financial NLP. Efficiency-accuracy tradeoffs are increasingly important, but statistically validated rank studies remain limited.

3 Methodology

3.1 Dataset

We use the **Twitter Financial News Sentiment** dataset from Hugging Face (`zeroshot/twitter-financial-n`
containing 9,543 labeled samples with three sentiment labels.

Table 1: Dataset Statistics

Split	**Samples**
Train	7,634
Test	1,909
Total	**9,543**
Labels: Bearish (0), Bullish (1), Neutral (2)	

3.2 Model

We use **DistilRoBERTa-base** (82M parameters), a distilled RoBERTa variant with 6 transformer layers and 768 hidden dimensions.

3.3 Multi-Seed Experimental Design

To reduce variance and support statistical testing, we run each configuration with **five random seeds**: [42, 123, 456, 789, 1337]. This affects:

- Weight initialization (classifier head and LoRA adapters)
- Dropout masks during training
- Data shuffling order

Total experiments: 6 configurations $\times$ 5 seeds = **30 runs**.

3.4 Experimental Configurations

Table 2: Experimental Configurations

Config	**Method**	**Trainable Params**	**% of Total**	**LoRA** α
1	Full Fine-Tuning	82,120,707	100.00%	—
2	LoRA $r = 4$	665,859	0.81%	8
3	LoRA $r = 8$	739,587	0.90%	16
4	LoRA $r = 16$	887,043	1.08%	32
5	LoRA $r = 32$	1,181,955	1.44%	64
6	LoRA $r = 64$	1,771,779	2.16%	128

LoRA Configuration:

- Target modules: `query`, `value` (attention layers)
- Dropout: 0.1
- Alpha scaling: $\alpha = 2r$
- Bias: None

3.5 Training Setup

Table 3: Training Hyperparameters

Parameter	Value
Batch Size	32
Learning Rate (Full FT)	2×10^{-5}
Learning Rate (LoRA)	1×10^{-4}
Epochs	3
Max Sequence Length	128
Optimizer	AdamW
Weight Decay	0.01
Precision	FP16 (mixed)

3.6 Statistical Analysis

We report mean ± standard deviation and 95% confidence intervals (CI) using the t-distribution. For pairwise comparisons, we use **paired t-tests** (same seed list across configurations) with significance threshold $\alpha = 0.05$.

4 Results

4.1 Multi-Seed Performance Summary

Table 4 reports accuracy and weighted F1 across all six configurations.

Table 4: Multi-Seed Experimental Results ($n = 5$ per configuration)

Configuration	Trainable (%)	Accuracy (mean±std)	95% CI	F1 (mean±std)
Full Fine-Tuning	100.00	**86.85±0.70%**	[85.98, 87.73]	86.86±0.69%
LoRA $r = 4$	0.81	82.20±0.83%	[81.17, 83.23]	82.12±0.89%
LoRA $r = 8$	0.90	83.55±0.55%	[82.87, 84.23]	83.57±0.53%
LoRA $r = 16$	1.08	84.08±0.61%	[83.32, 84.83]	84.11±0.58%
LoRA $r = 32$	1.44	84.42±0.89%	[83.31, 85.53]	84.49±0.90%
LoRA $r = 64$	2.16	<u>84.56±0.76%</u>	[83.62, 85.50]	84.65±0.73%

Bold = best overall; Underline = best among LoRA configurations

4.2 Key Finding: Diminishing Returns at $r = 16$

Table 5 shows paired t-tests for adjacent rank transitions.

Table 5: Statistical Significance of Rank Transitions (Paired t-tests)

Transition	**Δ Accuracy**	**t-statistic**	**p-value**	**Significant?**
$r = 4 \to r = 8$	+1.35 pp	5.798	0.0044	**Yes** **
$r = 8 \to r = 16$	+0.52 pp	8.209	0.0012	**Yes** **
$r = 16 \to r = 32$	+0.35 pp	1.855	0.1372	No
$r = 32 \to r = 64$	+0.14 pp	1.065	0.3469	No

Significance: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Interpretation: Accuracy improvements are statistically significant only up to $r = 16$. Beyond this point, gains are not distinguishable from random variance.

4.3 Full Fine-Tuning vs LoRA Comparisons

Table 6: Full Fine-Tuning vs Each LoRA Configuration (Paired t-tests)

Comparison	**Δ Accuracy**	**t-statistic**	**p-value**	**Significant?**
Full FT vs $r = 4$	+4.65 pp	31.124	<0.0001	**Yes** ***
Full FT vs $r = 8$	+3.30 pp	18.187	0.0001	**Yes** ***
Full FT vs $r = 16$	+2.78 pp	17.017	0.0001	**Yes** ***
Full FT vs $r = 32$	+2.43 pp	11.234	0.0004	**Yes** ***
Full FT vs $r = 64$	+2.29 pp	12.894	0.0002	**Yes** ***

All LoRA configurations perform significantly worse than full fine-tuning ($p < 0.001$), with gaps ranging from 2.29 pp ($r = 64$) to 4.65 pp ($r = 4$).

4.4 Run-to-Run Variance

Table 7: Run-to-Run Variance Analysis

Configuration	**Min**	**Max**	**Spread**	**Std Dev**
Full Fine-Tuning	85.96%	87.59%	1.62 pp	0.70%
LoRA $r = 4$	81.25%	83.34%	2.10 pp	0.83%
LoRA $r = 8$	82.77%	83.92%	1.15 pp	0.55%
LoRA $r = 16$	83.18%	84.70%	1.52 pp	0.61%
LoRA $r = 32$	83.18%	85.28%	2.10 pp	0.89%
LoRA $r = 64$	83.55%	85.28%	1.73 pp	0.76%

Typical run-to-run variance is approximately 1.5–2.0 percentage points. Single-run differences smaller than this are unreliable without multi-seed validation.

4.5 Visualization

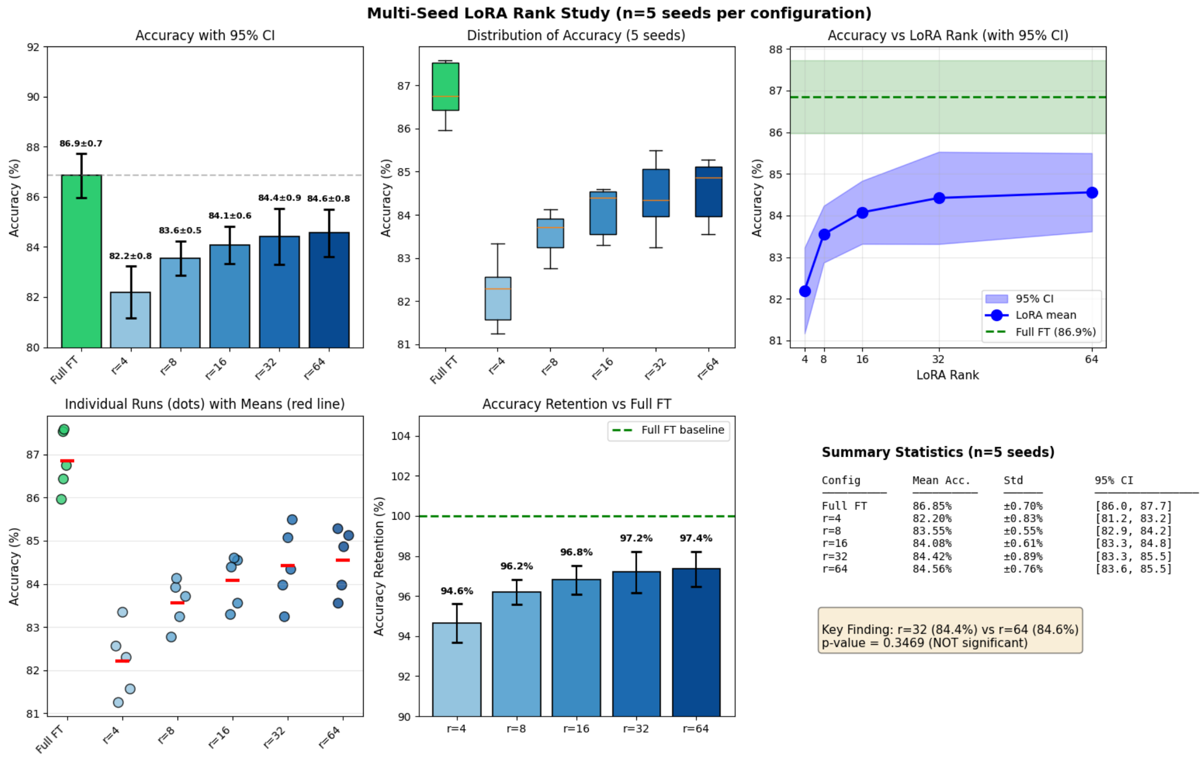

Figure 1: Multi-seed LoRA rank study results ($n = 5$ seeds per configuration). Key finding: $r = 32$ (84.4%) vs $r = 64$ (84.6%) is not statistically significant ($p = 0.35$).

4.6 Efficiency Analysis

Table 8: Efficiency Analysis with Statistical Validation

Config	Acc. Retention	Param Reduction	Gain vs Previous
LoRA $r = 4$	94.6%	99.2%	—
LoRA $r = 8$	96.2%	99.1%	Significant ($p = 0.004$)
LoRA $r = 16$	**96.8%**	98.9%	Significant ($p = 0.001$)
LoRA $r = 32$	97.2%	98.6%	Not significant ($p = 0.14$)
LoRA $r = 64$	97.4%	97.8%	Not significant ($p = 0.35$)

5 Discussion

5.1 Why Multi-Seed Validation Changes Conclusions

Single-run results can flip the apparent ranking between configurations. Multi-seed validation shows that $r = 32$ and $r = 64$ are statistically indistinguishable ($p = 0.35$), so claims that one is definitively better are not supported.

5.2 Interpretation: Intrinsic Rank Hypothesis

Results are consistent with the intrinsic-rank hypothesis in LoRA [Hu et al., 2022]: the task appears to have an effective adaptation rank around 16, beyond which additional capacity yields no statistically significant improvement.

5.3 Why Do Gains Plateau at $r = 16$?

We hypothesize:

1. **Task Complexity**: 3-class sentiment is a constrained downstream task.
2. **Dataset Size**: With roughly 7.6K training samples, higher ranks may not learn stable extra structure.
3. **Optimization Noise**: Additional degrees of freedom can amplify variance and reduce marginal gains.

5.4 Practical Deployment Recommendations

Table 9: Evidence-Based Deployment Recommendations

Use Case	Recommendation	Rationale
Maximum accuracy	Full Fine-Tuning	2.3–4.6 pp better ($p < 0.001$)
Production deployment	LoRA $r = 16$	Best within significant-gain zone
If rank cost is irrelevant	LoRA $r = 32$ or $r = 64$	Marginal, not significant
Memory-constrained	LoRA $r = 8$	Strong retention with fewer params
Rapid prototyping	LoRA $r = 4$	Fastest experiments

5.5 Limitations

1. **Single Dataset**: Rank sensitivity may vary across financial corpora.
2. **Single Model**: Larger models may exhibit different rank dynamics.
3. **Fixed Alpha Scaling**: We used $\alpha = 2r$.
4. **Seed Count**: $n = 5$ provides moderate power; $n = 10+$ would strengthen conclusions.

6 Conclusion

We present a multi-seed validated study of LoRA rank selection for financial sentiment analysis. Key findings:

1. **Diminishing returns at $r = 16$**: Significant gains occur only up to rank 16.
2. **$r = 32$ vs $r = 64$ is noise**: The 0.14 pp difference is not significant ($p = 0.35$).
3. **Multi-seed validation is essential**: Single-run differences under 2 pp are unreliable.
4. **Practical recommendation**: LoRA $r = 16$ is the best default for production efficiency.

6.1 Future Work

1. Increase seeds to $n = 10+$ for stronger statistical power.
2. Validate across Financial PhraseBank, SEC filings, and earnings transcripts.
3. Test larger backbones (Mistral-7B, Llama-3-8B) for rank sensitivity.
4. Explore automated rank selection methods for PEFT.

References

Dogu Araci. FinBERT: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

Chao Gao and Sai-Qian Zhang. DLoRA: Distributed parameter-efficient fine-tuning solution for large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Justin Zhao, Timothy Wang, Wael Abid, Geoffrey Angus, Arnav Garg, Jeffery Kinnison, Alex Sherstinsky, Piero Molino, Travis Addair, and Devvret Rishi. LoRA Land: 310 fine-tuned LLMs that rival GPT-4, a technical report. *arXiv preprint arXiv:2405.00732*, 2024.

A Individual Run Data

Table 10: All 30 Individual Run Results

Seed	**Full FT**	$r = 4$	$r = 8$	$r = 16$	$r = 32$	$r = 64$
42	87.53%	83.34%	83.92%	84.55%	85.49%	85.28%
123	85.96%	81.25%	83.24%	83.18%	83.18%	83.55%
456	86.43%	81.56%	82.77%	83.55%	84.13%	84.02%
789	86.75%	82.29%	83.87%	84.44%	84.86%	84.86%
1337	87.59%	82.56%	83.92%	84.70%	84.44%	85.12%
Mean	86.85%	82.20%	83.55%	84.08%	84.42%	84.56%
Std	0.70%	0.83%	0.55%	0.61%	0.89%	0.76%

B Reproducibility

Code: `lora_multiseed_experiment.ipynb`
Data: `multiseed_aggregated_results.json`
Hardware: NVIDIA GPU with CUDA support
Framework: PyTorch, Hugging Face Transformers, PEFT, SciPy

To reproduce:

1. Install dependencies: `pip install transformers datasets peft accelerate scipy`
2. Open notebook in Jupyter or Google Colab
3. Select GPU runtime
4. Run all cells sequentially

ValtricAI Research, December 2025