Parameter-Efficient Fine-Tuning of Qwen Language Models Using Quantized Low-Rank Adaptation (QLoRA)

Adib Akkari Tanim Chowdhury Omar El Masaoudi

Team: , COMP433 Course Project Proposal - Fall 2025

1. Problem Statement and Application

Open-weight Large Language Models (LLMs) like Qwen face a critical challenge: full fine-tuning requires prohibitive computational resources (hundreds of GBs of VRAM), making specialized model adaptation inaccessible to researchers with limited hardware. This represents the *LLM Adaptation Bottleneck*.

Our project addresses: *How can we efficiently adapt LLMs to downstream tasks while minimizing computational requirements without sacrificing performance?* We tackle this by reproducing Quantized Low-Rank Adaptation (QLoRA) [1], which combines 4-bit quantization with low-rank adapters to drastically reduce memory and trainable parameters.

Goals and Expectations: We aim to demonstrate that QLoRA enables effective fine-tuning of models up to 8B parameters on a single 24GB consumer GPU (RTX 3090), achieving competitive performance with full fine-tuning while reducing trainable parameters by over 99%. This validates QLoRA as a state-of-the-art solution for accessible LLM adaptation.

2. Reading Material

Our work is grounded in the following literature:

1. **Dettmers et al. (2023)** [1]: QLoRA: Efficient Finetuning of Quantized LLMs (NeurIPS 2023). *Primary SOTA method for re-implementation.*
2. **Hu et al. (2021)** [2]: LoRA: Low-Rank Adaptation of Large Language Models (ICLR 2022).
3. **Zhang et al. (2024)** [3]: Qwen2.5 Technical Report. Provides model architecture specifications.
4. **Houlsby et al. (2019)** [4]: Parameter-Efficient Transfer Learning for NLP (ICML 2019).

3. Possible Methodology

We will re-implement QLoRA using the Hugging Face PEFT library and `bitsandbytes` for quantization. Our implementation systematically explores different model scales and task configurations.

Hardware: NVIDIA RTX 3090 (24GB VRAM), 32GB DDR5 RAM.

Models: Qwen2.5-4B-Instruct and Qwen2.5-8B-Instruct from Hugging Face.

Configuration: 4-bit NormalFloat (NF4) quantization for frozen base weights, `bfloat16` precision for LoRA adapters, Paged AdamW optimizer with gradient checkpointing.

QLoRA Parameters: Rank $r \in \{4, 8, 16\}$ (systematic ablation), LoRA Alpha $\alpha = 16$, target modules include Query, Key, Value projections in attention layers.

Datasets: (1) SST-2 [5] for sentiment classification, (2) SQuAD v1.1 [6] for extractive question answering, (3) AlpacaEval [7] for instruction following.

Improvements and Extensions: We extend the original QLoRA work through: (1) Scaling analysis comparing 4B vs 8B models, (2) Systematic rank ablation study, (3) Adapter placement comparison (attention-only vs attention+MLP), (4) Efficiency benchmarking against full fine-tuning baselines.

4. Metric Evaluation

Quantitative Analysis:

- **Performance Metrics:** Accuracy (SST-2), F1 Score and Exact Match (SQuAD), Win Rate (AlpacaEval)
- **Efficiency Metrics:** GPU memory usage (GB), trainable parameter count and percentage, training throughput (tokens/sec)

Qualitative Analysis:

- Plots: Rank vs. Accuracy trade-off curves across tasks
- Figures: Scaling efficiency (Performance vs. VRAM) for 4B and 8B models
- Visualizations: Validation loss curves comparing QLoRA and full fine-tuning
- Examples: Side-by-side generation comparisons for qualitative assessment

We expect QLoRA to achieve within 3% of full fine-tuning performance while using less than 1% trainable parameters and fitting the 8B model within our 24GB VRAM budget.

References

[1] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1

[2] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *International Conference on Learning Representations (ICLR)*, 2022. 1

[3] A. Zhang et al., "Qwen2.5: A Party of Foundation Models," Alibaba Group Technical Report, 2024. 1

[4] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-Efficient Transfer Learning for NLP," in *International Conference on Machine Learning (ICML)*, 2019. 1

[5] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," in *Proceedings of EMNLP*, 2013. 1

[6] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in *Proceedings of EMNLP*, 2016. 1

[7] "AlpacaEval: An Automatic Evaluator for Instruction-following Language Models," https://github.com/tatsu-lab/alpaca_eval, 2023. 1

Gantt Chart (Supplemental Material)

Table 1. 8-Week Project Development Schedule

Week	Phase	Milestones & Deliverables
1	Setup	Literature review, environment setup, dataset preprocessing
2	Baseline	QLoRA on Qwen2.5-4B + SST-2 ($r = 8$) working
3	Scaling	Extend to Qwen2.5-8B, SST-2 experiments complete
4	Multi-Task	SQuAD v1.1 experiments for both models
5	Ablations	Rank ablation ($r = 4, 8, 16$), adapter placement tests
6	Final Exps	AlpacaEval experiments, efficiency benchmarking
7	Analysis	Compute metrics, generate visualizations, begin report
8	Finalization	Complete report, code documentation, presentation