Base Model Is All You Need

Chuan Chen, Jiajun Qi, Rui Wang, Yifan Xu

Johns Hopkins Whiting School of Engineering

Introduction

As large language models (LLMs) continue to grow in size and capability, adapting them to downstream tasks under **realistic hardware constraints** has become increasingly important. This project focuses on **task-specific fine-tuning** of LLMs while ensuring that the resulting models can run efficiently on a **20GB GPU**. We explore several **parameter-efficient fine-tuning** and **knowledge distillation** techniques across different models, evaluating each using **perplexity** on a 2000-sentence test set.

Our **best result**, achieved with **LoRA fine-tuning**, reaches **perplexity of 5.9986** on the test set**.**

The Problem

Goal: Fine-tune a language model to achieve low perplexity on a given test dataset.

Constraints:

- **Limited GPU Memory** (20GB MIG GPU) for inference
- **Limited Provided Data**: 2k-sentence training set + 2k-sentence test set

The key challenge is to apply **efficient fine-tuning techniques** that improve model performance **without exceeding hardware limitations**.

Methods

- **Model Benchmarking** – Evaluated baseline performance across multiple LLMs.
- **LoRA & QLoRA** – Parameter-efficient tuning under memory constraints.
- **Prefix Tuning** – Tuned soft prompts by varying virtual token length.
- **Full Finetuning** – Applied to models with ≤ 2B parameters.
- **Distillation** – Transferred knowledge from larger to smaller models.
- **Data Augmentation** – Enlarged training dataset size to improve generalization.
- **Quantization** – Used BF16/FP16 to inference within 20GB GPU memory.

Outcomes and Results

1. Model Size vs. Memory Feasibility

To meet the 20GB GPU constraint, we tested model sizes under different precisions:

- **Models ≤ 2B** run comfortably in **FP32**, using 5–13GB of memory.
- **7B/8B models' inference** will exceed 20GB in **FP32**, but fit within 17–19GB using **BF16** or **FP16**.

Accordingly, we used **FP32** for small models and **BF16/FP16** for larger ones.

2. Baseline Model Evaluation

2.1 Cross-Family Baseline Comparison

To ensure a fair comparison under the 20GB GPU constraint, we set **Max Length** of all model inputs to **2048 tokens**—matching **TinyLlama**'s maximum context length and covering most of our dataset.

Granite 3.3 and **TinyLlama Chat-v1.0** achieved the best baseline perplexities, so we selected **Granite** series and **TinyLlama** series models for further fine-tuning.

Model	PPL	Max Length	Precision	Params
TinyLlama-1.1B-Chat-v1.0	**8.122**	2048	FP32	1.1B
TinyLlama-1.1B-Chat-v1.0	**8.122**	2048	BF16	1.1B
Qwen2.5-0.5B	15.098	4096	FP32	0.5B
Qwen2.5-0.5B	15.104	4096	BF16	0.5B
Qwen2.5-0.5B-Instruct	16.562	2048	FP16	0.5B
Qwen2.5-1.5B	11.831	2048	FP32	1.5B
Qwen2.5-1.5B-Instruct	11.834	2048	FP32	1.5B
Qwen2.5-7B-Instruct	9.602	2048	BF16	7B
Qwen2.5-7B-Instruct-1M	9.666	2048	BF16	7B
Qwen3-4B	14.924	2048	BF16	4B
DeepSeek-R1-Distill-Qwen-1.5B	36.525	2048	FP32	1.5B
DeepSeek-R1-Distill-Qwen-1.5B	36.578	2048	BF16	1.5B
DeepSeek-llm-7b-chat	9.274	2048	BF16	7B
Llama-3.2-1B-Instruct	15.978	2048	FP32	1B
Llama-3.2-3B-Instruct	12.454	2048	FP32	3B
granite-3.3-2b-base	**6.943**	2048	FP32	2B
granite-3.3-2b-instruct	8.941	2048	FP32	2B
granite-3.3-2b-instruct	8.951	2048	BF16	2B
granite-3.3-8b-base	**6.118**	2048	BF16	8B

2.2 Intra-Family Variant Comparison (Granite)

To explore performance scaling, we conducted an intra-family comparison across **Granite**-3.0 to 3.3 series, as Granite offers multiple variants.

The best-performing variants, **granite-3.0-8b-base** and **granite-3.3-8b-base**, were selected for subsequent fine-tuning experiments.

Model	3.0	3.1	3.2	3.3
2b-base	6.926	7.608	N/A	6.949
2b-instruct	9.722	9.167	9.196	8.951
8b-base	**6.075**	6.659	N/A	**6.118**
8b-instruct	6.800	6.726	6.770	7.255

3. Results of Fine-Tuning Strategies

3.1 LoRA and QLoRA

LoRA consistently outperformed QLoRA in perplexity across models. Its 16-bit precision preserves learning signals more effectively on small datasets, while QLoRA's 4-bit quantization tends to lose information.

Models	Baseline PPL	LoRA PPL	QLoRA PPL
TinyLlama-1.1B-Chat-v1.0	8.122	8.184	8.425
qwen2.5-0.5B	15.133	15.49	17.093
Qwen2.5-1.5B-Instruct	11.845	11.684	12.415
Llama-3-2-1B-Instruct	15.978	15.046	15.966
Llama-3-2-3B-Instruct	12.454	11.6861	12.117
granite-3.2-2b-instruct	9.196	7.251	7.525
granite-3.3-2b-instruct	8.951	7.658	7.922
granite-3.0-8b-base	**6.075**	**6.019**	**6.209**
granite-3.3-8b-base	6.118	**6.019**	6.214

To further improve LoRA performance, we conducted a grid search over **rank**, **α**, and **dropout** on granite-3.0-8b-base, our best-performing variant.

Results showed that **LoRA is sensitive to these hyperparameters**.

The best configuration achieved a **perplexity of 5.9986**, the lowest among all our experiments.

PPL	Rank	Lora Alpha	Dropout	Best Epoch
6.0025	8	16	0.05	2
6.0080	4	8	0.01	2
5.9987	8	32	0.05	1
5.9986	16	32	0.05	2
5.9998	16	64	0.05	2

3.2 Prefix Finetuning

We conducted prefix tuning experiments to evaluate the impact of different prefix lengths and learning rates. The best result (PPL = 6.027) was achieved by granite-3.3-8b-base with a prefix length of 2/4 and a learning rate of 1e-4.

Base Model	Prefix Length	Learning Rate	Baseline PPL	PPL
granite-3.0-8b-base	4	1e-4	6.075	6.029
granite-3.0-8b-base	8	1e-4	6.075	6.052
granite-3.1-8b-base	4	1e-4	6.659	6.042
granite-3.3-2b-base	8	1e-4	6.949	6.959
granite-3.3-8b-base	2	1e-4	6.118	**6.027**
granite-3.3-8b-base	4	1e-4	6.118	**6.027**
granite-3.3-8b-base	4	2e-4	6.118	6.029
granite-3.3-8b-base	8	1e-4	6.118	6.038
granite-3.3-8b-base	16	1e-4	6.118	6.062

3.3 Full Finetuning Evaluation and Distillation

We fine-tuned granite-3.3-2b-base, improving perplexity from 6.949 → 6.88, outperforming TinyLlama-1.1B (8.122 → 7.7). Believing full fine-tuning is effective for 2B models, we combined it with distillation from granite-3.3-8b-LoRA, hoping the student could benefit from soft supervision.

However, top-10 and top-100 token-level distillation led to worse results (7.756, 7.057), suggesting some problems in training.

We plan to further analyze this in future work.

Model/Condition	Original PPL	Final PPL
granite-3.3-2b-base	6.949	6.883
TinyLlama-1.1B-Chat-v1.0	8.122	7.745
granite-3.3-2b-base distill top-10	6.949	7.756
granite-3.3-2b-base distill top-100	6.949	7.057

3.4 Data Augmentation

We applied several augmentation methods to expand the training set by 1.5 times, including:

- Synonym replacement, word swap, insertion, and deletion (EDA)
- Back-translation via English ↔ German
- Random masking

Despite this effort, perplexity **increased significantly**, showing that augmentation hurt performance rather than helped. We suspect this is due to **distribution mismatch**: the original training and test sets may have been generated by the same large model and share hidden patterns. In contrast, our augmented samples broke this structure, introducing noise and reducing generalization.

This suggests that **naive augmentation is ineffective in this setting.**

Conclusion

- **16-bit Quantization** enabled inference of larger 8B models within the 20GB GPU.
- **Full fine-tuning** reduced perplexity but showed limited gains due to data scarcity. Combined with distillation, performance unexpectedly degraded. We plan to revisit this approach in future work.
- **Prefix fine-tuning** results varied with prefix length. Shorter prefixes performed better, likely due to limited data. Optimal configurations also differed across model families.
- **Data augmentation** significantly worsened performance, likely because it disrupted the original data distribution.
- **LoRA consistently outperformed QLoRA**, suggesting that when data is limited, preserving signal precision is more important than further parameter reduction.
- Our **best model**, **granite-3.0-8b-base** with LoRA, achieved a **perplexity of 5.9986** on the test set.