Benchmarking Parameter-Efficient Fine-Tuning for Transformer Models

Zhejun Yu, Zixu Geng, Lixun Zhang

Introduction

Large language models (LLMs) achieve state-of-the-art performance but require heavy computation and memory during fine-tuning. Traditional full-model fine-tuning updates all parameters, making it expensive and impractical.

Parameter-Efficient Fine-Tuning (PEFT) offers a solution by reducing trainable parameters while maintaining competitive accuracy.

Methodology

PEFT techniques modify only a small subset of parameters or introduce lightweight trainable modules, enabling efficient adaptation of large models.

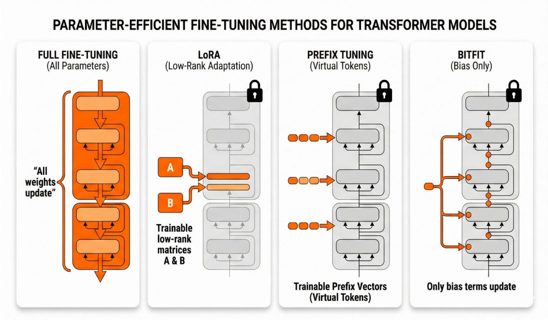

Figure 1: Overview of parameter-efficient fine-tuning methods for transformer models

We focus on three influential PEFT methods:

- LoRA: Injects trainable low-rank matrices into selected linear projections while keeping the backbone frozen.
- Prefix Tuning: Prepends a small set of trainable virtual tokens to each layer's attention mechanism without modifying model weights.
- BitFit: Updates only the bias terms across layers while freezing all weight matrices.

Our baseline is full fine-tuning, allowing direct comparison of cost and performance.

Experimental Evaluation

- Backbone models: T5-small, BERT-base
- Tasks: SST-2, MRPC
- Compared methods: Full FT, LoRA, Prefix, BitFit

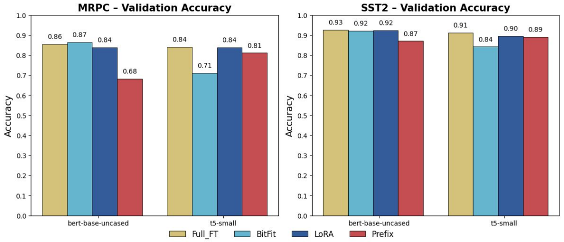

Figure 2: Validation Accuracy Comparison Across PEFT Methods and Models

Across both MRPC and SST-2, parameter-efficient fine-tuning (PEFT) methods achieve performance close to full fine-tuning. LoRA consistently yields the best accuracy–efficiency trade-off, reaching **0.84–0.92** accuracy compared to 0.86–0.93 under full fine-tuning. BitFit updates only **0.1%** of parameters and remains competitive, while Prefix shows slightly lower accuracy. All PEFT methods reduce GPU memory usage by up to **55%** and substantially shorten training time.

Table 1: Efficiency–Performance Trade-off of PEFT Methods

Task	Model	Method	Accuracy	F1 Score	Trainable Params (%)	GPU Mem (MB)	Train Time (s)
MRPC	BERT	Full_FT	0.855	0.898	100.00	2346	113
		BitFit	**0.865**	**0.904**	**0.10**	**1135**	**85**
		LoRA	0.838	0.884	0.41	1354	102
		Prefix	0.681	0.810	0.34	1162	**85**
	T5	Full_FT	**0.841**	**0.888**	100.00	2202	119
		BitFit	0.711	0.822	0.43	1195	**102**
		LoRA	0.838	0.885	0.96	1590	183
		Prefix	0.811	0.865	**0.20**	**358**	103
SST-2	BERT	Full_FT	**0.925**	**0.928**	100.00	1933	1273
		BitFit	0.921	0.923	**0.10**	**866**	**1013**
		LoRA	0.924	0.927	0.41	1012	1653
		Prefix	0.870	0.874	0.34	908	1035
	T5	Full_FT	**0.912**	**0.915**	100.00	1617	2030
		BitFit	0.842	0.846	0.43	804	1724
		LoRA	0.896	0.898	0.96	1059	3180
		Prefix	0.891	0.895	**0.20**	**308**	**1439**

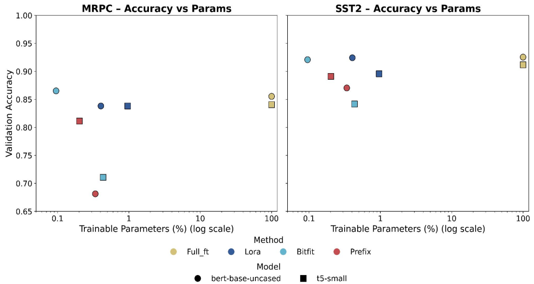

Figure 3: Validation Accuracy vs. Trainable Parameters on MRPC and SST-2

Conclusion

Across MRPC and SST-2, PEFT methods match full fine-tuning performance while reducing trainable parameters by over 99%. LoRA provides the strongest balance of accuracy and efficiency, maintaining near–full performance with much lower memory and compute cost. Prefix Tuning offers substantial parameter savings with moderate performance drops, while BitFit shows greater variability across tasks.