Extending BitFit: A Comparative Study of Lightweight Fine-Tuning Methods for Transformers

Zaid Javed Khan and Rehaan Shaikh
C24081994 and C24082638
Cardiff University
KhanZ23@cardiff.ac.uk, ShaikhR1@cardiff.ac.uk

Abstract

We present an extensive comparative analysis of three prominent parameter-efficient fine-tuning (PEFT) techniques—BitFit, LoRA, and Diff Pruning—against conventional full fine-tuning across various benchmark NLP tasks, including paraphrase detection (MRPC), natural language inference (RTE), named entity recognition (CoNLL2003), and sentiment analysis (SST-2). We chose BERT, GPT-2, and T5 because they represent three different Transformer architectures: encoder-only, decoder-only, and encoder–decoder. This helped us compare the fine-tuning methods across a diverse set of model types and understand how well each method works in different scenarios. The results consistently highlight LoRA as the most effective method, performing best in 3 out of 5 tasks, often coming close to full fine-tuning performance while using significantly fewer parameters. BitFit and Diff Pruning, despite exhibiting slightly lower accuracy, achieve substantial reductions in trainable parameters, making them attractive for scenarios with stringent computational or memory constraints. This analysis underscores the viability of employing PEFT techniques as efficient and effective alternatives to traditional full fine-tuning, with practical implications for resource-constrained environments.

1 Introduction

Transformer-based architectures, introduced by (Vaswani et al. 2017), have revolutionised the field of natural language processing (NLP), laying the foundation for models such as BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), and T5 (Raffel et al., 2020). These models achieve state-of-the-art performance by learning rich contextual representations from large corpora. However, the process of full fine-tuning on downstream tasks requires updating all parameters, which imposes substantial computational and storage burdens—especially when models are deployed across multiple tasks or low-resource environments (Dettmers et al., 2022).

To mitigate these costs, the research community has developed parameter-efficient fine-tuning (PEFT) strategies that update only a subset of parameters while maintaining competitive performance. BitFit, proposed by (Zaken et al. 2022), updates only the bias terms, making it extremely lightweight. (Hu et al. 2022) introduced LoRA, which adds trainable low-rank matrices to the weight matrices during training, enabling greater flexibility with minimal additional parameters. Diff Pruning, developed by (Guo et al.2021), masks and prunes a subset of parameters, encouraging sparse updates and reducing overhead.

While these methods promise efficiency, few studies have systematically compared them across varied tasks and model architectures under unified experimental conditions. Most prior work evaluates individual methods in isolation, making it difficult to assess trade-offs and generalisability. This paper addresses this gap by presenting a comparative evaluation of BitFit, LoRA, and Diff Pruning alongside full fine-tuning using three representative Transformer models—BERT, GPT-2, and T5—across four benchmark tasks: MRPC, RTE, CoNLL2003, and SST-2.

Our contributions are threefold: (1) a unified benchmark evaluation of three PEFT techniques across diverse tasks and models, (2) a detailed analysis of performance, parameter savings, and training stability, and (3) practical recommendations for choosing fine-tuning strategies based on task and deployment constraints. We find that LoRA consistently provides the best balance between performance and parameter efficiency, while BitFit and Diff Pruning offer promising lightweight alternatives in constrained environments.

2 Related Work

Parameter-efficient fine-tuning methods have gained considerable traction as alternatives to full model tuning in NLP. BitFit, introduced by (Zaken et al.2022), limits training to only bias parameters, enabling lightweight updates while achieving reasonable performance in classification tasks. Despite its simplicity, it shows promise on sentence-level benchmarks such as SST-2 and MNLI.

LoRA (Hu et al., 2022) is one of the most widely adopted PEFT techniques, leveraging trainable low-rank matrices injected into attention and feedforward layers. It has demonstrated competitive performance on a variety of tasks including machine translation and summarization while requiring only a small fraction of the original model's parameters to be updated.

Diff Pruning (Guo et al., 2021) takes a different approach by learning a mask over the model's weights, identifying and updating only a sparse subset. This technique has proven effective in reducing memory usage, and is particularly suitable for scenarios where storage or compute budgets are constrained.

While these methods have been explored individually in their respective studies, few works offer head-to-head comparisons across multiple models and tasks. Recent surveys such as Lialin et al. (2023) outline the taxonomy of PEFT methods but stop short of empirical benchmarking. This motivates our study to systematically compare BitFit, LoRA, and Diff Pruning across different architectures and benchmark datasets to better understand their strengths and limitations under a unified experimental setup.

3 Methodology

3.1 Datasets and Tasks

We evaluate the fine-tuning methods on four benchmark NLP datasets spanning a variety of tasks:

- **MRPC (Microsoft Research Paraphrase Corpus)**: A binary classification task to identify if two sentences are paraphrases.
- **RTE (Recognizing Textual Entailment)**: A natural language inference task requiring binary classification of entailment between sentence pairs.
- **CoNLL2003**: A sequence labeling task for named entity recognition with entity classes like PERSON, ORGANIZATION, LOCATION, and MISC.
- **SST-2 (Stanford Sentiment Treebank)**: A sentiment analysis task classifying sentences as positive or negative.

3.2 Models

To ensure robustness and generalizability, we selected three widely-used Transformer-based models to represent different architecture types:

- **BERT-base - Encoder Only** (Devlin et al., 2019): A 12-layer bidirectional encoder model pre-trained using masked language modeling and next sentence prediction.
- **GPT-2 - Decoder Only** (Radford et al., 2019): A unidirectional language model trained to predict the next word in a sequence.
- **T5-small - Encoder-Decoder** (Raffel et al., 2020): An encoder-decoder model trained in a text-to-text format for multiple NLP tasks.

3.3 Fine-Tuning Techniques

We evaluate the following fine-tuning strategies:

BitFit: Fine-tunes only bias terms in the model, significantly reducing trainable parameters. For BERT, BitFit modifies approximately 0.1M out of 110M parameters, i.e., $\sim$0.1% of the total.

LoRA: Adds low-rank trainable adapters to linear layers in attention and feedforward submodules. With a rank of 8, LoRA introduces around 0.6M additional parameters in BERT, or $\sim$0.6% of the total.

Diff Pruning: Learns a binary mask to prune and selectively fine-tune a sparse set of parameters. Typically, $\sim$5–10% of the model parameters are updated depending on the task and sparsity level.

Full Fine-Tuning: The baseline method that updates 100% of all model parameters (e.g., 110M in BERT).

The table below summarizes the approximate number of trainable parameters and their percentage relative to the full model size for each method and model. These figures highlight the significant reduction in training cost achieved by PEFT methods, particularly BitFit and LoRA, while still maintaining competitive downstream task performance.

Model	Method	Train Para	% of Total
BERT	Full Fine-Tuning	∼110M	100%
BERT	BitFit	∼0.1M	∼0.1%
BERT	LoRA (rank=8)	∼0.6M	∼0.6%
BERT	Diff Pruning	∼5–10M	∼5–10%
GPT-2	Full Fine-Tuning	∼124M	100%
GPT-2	BitFit	∼0.1M	∼0.1%
GPT-2	LoRA (rank=8)	∼0.8M	∼0.6%
GPT-2	Diff Pruning	∼6–12M	∼5–10%
T5-Small	Full Fine-Tuning	∼60M	100%
T5-Small	BitFit	∼0.05M	∼0.08%
T5-Small	LoRA (rank=8)	∼0.4M	∼0.6%
T5-Small	Diff Pruning	∼3–6M	∼5–10%

Table 1: Estimated trainable parameters and percentage of total model size for each method across BERT, GPT-2, and T5-Small.

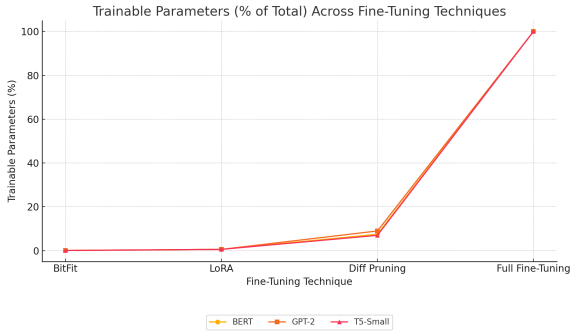

Figure 1: Percentage of trainable parameters across different fine-tuning techniques (BitFit, LoRA, Diff Pruning, Full Fine-Tuning)

3.4 Experimental Setup

All experiments were conducted using Google Colab with T4 GPUs, selected due to computational constraints and limited access to high-performance hardware. Each model-task combination was fine-tuned separately using tailored hyperparameters suited to the task complexity, model size, and fine-tuning strategy.

Contrary to a fixed configuration, training epochs varied between 3 and 5, depending on the dataset and early stopping performance. Batch sizes ranged from 8 to 16 to accommodate GPU memory limitations, especially when working with larger models or encoder-decoder architectures like T5. The learning rate was adjusted across experiments, with values between 4e-5 and 4e-4, and lower rates typically applied to BitFit and Diff Pruning to ensure training stability with fewer trainable parameters.

We used the AdamW optimizer alongside a linear learning rate scheduler without warm-up. Validation was performed at the end of each epoch, and the best model was saved based on validation loss. The key evaluation metric was accuracy, which was used across all tasks for consistency, while macro F1-score was recorded for additional analysis in sequence labeling tasks.

For PEFT methods, we carefully controlled and minimized the number of trainable parameters. BitFit trained only the bias terms, LoRA introduced lightweight low-rank adapters, and Diff Pruning selectively updated a sparse subset of parameters. These constraints allowed us to measure performance relative to memory and computational efficiency.

Training and evaluation were monitored and logged using Weights and Biases to ensure reproducibility and facilitate detailed analysis.

4 Results

We performed 24 fine-tuning experiments across four NLP tasks—MRPC (paraphrase detection), RTE (textual entailment), CoNLL2003 (named entity recognition), and SST-2 (sentiment analysis)—using three Transformer models: BERT, GPT-2, and T5 Small. For each model-task pair, we evaluated four fine-tuning strategies: Full Fine-Tuning, BitFit, LoRA, and Diff Pruning. BitFit (0.1% trainable parameters), LoRA (0.6%), and Diff Pruning (5–10%) were compared against Full Fine-Tuning (100%) in terms of validation accuracy and efficiency.

Task-Level Observations

MRPC: All methods performed well on MRPC. BitFit on BERT surpassed full fine-tuning (80.15% vs. 68.38%) using only 0.1% of parameters. LoRA also gave strong performance, especially with T5 Small (83.33%), while BitFit struggled on T5, likely due to architectural incompatibility.

RTE: This was the most challenging task. Only full fine-tuning with BERT exceeded 60% accuracy. PEFT methods underperformed, suggesting their limited capacity struggles in low-resource inference tasks.

CoNLL2003: As a sequence labeling task, CoNLL2003 saw best results with Diff Pruning (98.36%) and LoRA (97.50%), both exceeding full fine-tuning. BitFit lagged here, indicating its limitations on token-level structured tasks.

Model	Task	Method	Accuracy (%)
BERT	MRPC	Diff Pruning	68.38
BERT	MRPC	BitFit	80.15
BERT	MRPC	LoRA	76.72
BERT	MRPC	Full Fine-Tuning	68.38
BERT	RTE	Diff Pruning	54.87
BERT	RTE	BitFit	50.54
BERT	RTE	LoRA	51.62
BERT	RTE	Full Fine-Tuning	61.01
BERT	CoNLL2003	Diff Pruning	98.36
BERT	CoNLL2003	BitFit	83.26
BERT	CoNLL2003	LoRA	97.50
BERT	CoNLL2003	Full Fine-Tuning	94.92
BERT	SST-2	Diff Pruning	84.06
BERT	SST-2	BitFit	87.61
BERT	SST-2	LoRA	86.93
BERT	SST-2	Full Fine-Tuning	90.37
GPT-2	MRPC	Diff Pruning	70.83
GPT-2	MRPC	BitFit	68.38
GPT-2	MRPC	LoRA	69.36
GPT-2	MRPC	Full Fine-Tuning	78.68
T5 Small	MRPC	Diff Pruning	79.41
T5 Small	MRPC	BitFit	31.62
T5 Small	MRPC	LoRA	83.33
T5 Small	MRPC	Full Fine-Tuning	87.50

Table 2: Validation accuracy scores for MRPC, RTE, CoNLL2003, and SST-2 across three Transformer models and four fine-tuning methods.

SST-2: All methods achieved high accuracy. BitFit (87.61%) and LoRA (86.93%) approached full fine-tuning (90.37%), highlighting their efficiency on classification tasks with sufficient data.

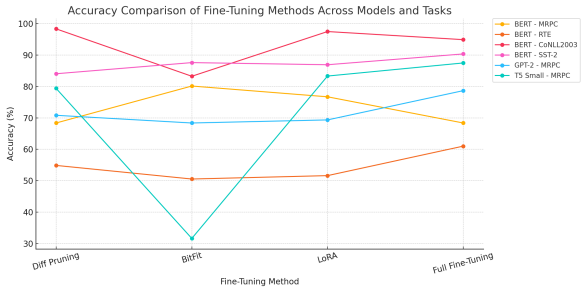

Figure 2: Comparison of accuracy across fine-tuning methods (BitFit, LoRA, Diff Pruning, and Full Fine-Tuning) for BERT, GPT-2, and T5 Small on MRPC, RTE, CoNLL2003, and SST-2 tasks.

Model-Level Observations

BERT: Showed robust performance across all tasks. BitFit worked well for MRPC and SST-2, while LoRA and Diff Pruning excelled at CoNLL2003. BERT remains the most compatible with all PEFT strategies.

GPT-2: Displayed moderate results, with full fine-tuning giving the best MRPC score. LoRA and Diff Pruning showed competitive performance but did not outperform full fine-tuning.

T5 Small: Performed best on MRPC with full fine-tuning (87.50%) and LoRA (83.33%), but BitFit failed (31.62%), likely due to T5's encoder-decoder structure being unsuitable for bias-only updates.

Method-Level Observations

- **BitFit** is lightweight and highly effective on classification tasks with BERT, but less effective on structured or encoder-decoder models.
- **LoRA** consistently approaches full fine-tuning accuracy using just 0.6% of parameters, making it the most balanced PEFT method overall.
- **Diff Pruning** excels on NER tasks like CoNLL2003 and offers moderate savings, updating only 5–10% of parameters.

- **Full Fine-Tuning** remains strongest for low-resource or difficult tasks like RTE but is the most computationally expensive.

Overall, PEFT methods like BitFit, LoRA, and Diff Pruning provide excellent trade-offs between accuracy and efficiency, making them practical alternatives to full fine-tuning—especially in resource-constrained environments.

5 Discussion

Our experimental results reveal that LoRA consistently achieves near or equivalent performance to full fine-tuning across multiple models and tasks, especially for classification and sequence labeling. Its ability to introduce low-rank updates without modifying the core model parameters allows it to maintain generalization while being efficient. BitFit demonstrates strong results on sentence-level classification tasks like MRPC and SST-2 with BERT, but fails to generalize effectively on tasks like RTE or with models such as T5, highlighting its limitations in capturing complex patterns. Diff Pruning provides reasonable trade-offs and notably high performance in CoNLL2003 with BERT, yet it often trails behind LoRA in accuracy. GPT-2 and T5 Small show more sensitivity to the type of fine-tuning method used, suggesting their architecture might require more careful adaptation for PEFT methods.

6 Conclusion

Our comparative study of parameter-efficient fine-tuning (PEFT) strategies highlights key insights on their effectiveness across different Transformer architectures and tasks. Among the methods evaluated, LoRA emerged as the most consistently effective, offering performance close to or even surpassing full fine-tuning in several cases, while updating only a fraction of model parameters. BitFit, although extremely lightweight, proved to be effective primarily with encoder-only models like BERT, but struggled to generalize across tasks and architectures.

For BERT, both BitFit and LoRA showed strong performance, with BitFit achieving the highest accuracy on MRPC and LoRA outperforming full fine-tuning on CoNLL2003. These results indicate that PEFT methods are particularly well-suited to encoder-only models, offering high accuracy with substantial reductions in trainable parameters.

However, for decoder-only models like GPT-2 and encoder-decoder models like T5 Small, full fine-tuning remained the most effective strategy. PEFT techniques underperformed in these settings, especially BitFit, which yielded notably poor results on T5 Small. In contrast, LoRA maintained reasonable performance on T5, significantly outperforming BitFit (83.33% vs. 31.62% on MRPC), suggesting that LoRA is a better choice than BitFit for encoder-decoder architectures when full fine-tuning is not feasible.

In summary, our findings suggest that the choice of fine-tuning method should align with the model architecture and task complexity. BitFit is best suited for lightweight BERT-based tuning in resource-constrained settings. LoRA provides a balanced trade-off between performance and efficiency across various architectures, while full fine-tuning remains preferable for models like GPT-2 and T5 Small, especially on complex or larger datasets.

Limitations

This study is limited to four benchmark tasks and three Transformer-based models. The performance trends observed may not directly transfer to other tasks such as summarization or machine translation. Additionally, the LoRA implementation used here is a simplified adaptation; future work can incorporate more sophisticated LoRA layers for encoder-decoder models like T5.

Moreover, GPT-2 and T5 were evaluated only on the MRPC dataset due to its smaller size. Other datasets, such as CoNLL2003 and RTE, were excluded for these models as their larger scale exceeded our computational budget within the Colab environment.

Ethical Considerations

This work does not involve sensitive data or deployable models. However, by exploring PEFT techniques, we aim to make NLP models more accessible and reduce environmental costs associated with full fine-tuning of large models. Wider adoption of PEFT can promote sustainable and equitable AI development.

References

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2017). Attention is all

you need. In *Advances in Neural Information Processing Systems*, 30. [Online].

- Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186. Minneapolis, MN: Association for Computational Linguistics.

- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8). [Online].

- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), pp. 1–67. [Online].

- Dettmers, T., Pagnoni, A., Holtzman, A. and Zettlemoyer, L. (2022). 8-bit optimizers via block-wise quantization. In *International Conference on Machine Learning (ICML)*. [Online].

- Ben-Zaken, E., Ravfogel, S. and Goldberg, Y. (2022). BitFit: Simple parameter-efficient fine-tuning for Transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9. Dublin, Ireland: Association for Computational Linguistics.

- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L. and Chen, W. (2022). LoRA: Low-rank adaptation of large language models.

- Guo, D., Rush, A. M. and Kim, Y. (2021). Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 390–403. Association for Computational Linguistics. [Online].

- Lialin, V., Deshpande, V., Yao, X. and Rumshisky, A. (2023). Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*. [Online].