Global Social Science and Humanities Journal
Online :ISSN 2964-1543
gsshj.globaledupublisher.com
Global Education Publisher

Balancing Efficiency and Performance: A Comparative Study of LoRA, Adapters, and Prompt Tuning

Wu Lingyi[1*], Anwar Saif[2]

[1]Guangdong Technology College, China, email: 76453501@qq.com

[2]Department of Information Systems, Sana'a University, Sana'a, Yemen, email: anwarsaif@su.edu.ye

Corresponding Author email: 76453501@qq.com

Received: Mar 11, 2026
Revised : Apr 2, 2026
Accepted: Apr 25, 2026
Published : May 04, 2026

Abstract: The rapid growth of pre-trained language models has substantially improved performance across Natural Language Processing (NLP) tasks. However, full fine-tuning remains computationally expensive because it requires updating and storing all model parameters for each downstream task. This limitation is particularly important in resource-constrained environments where access to high-performance computing infrastructure is limited. Parameter-efficient fine-tuning (PEFT) methods, including Low-Rank Adaptation (LoRA), Adapters, and Prompt Tuning, offer practical alternatives by updating only a small subset of parameters while keeping most of the pre-trained model frozen. This study presents a controlled and reproducible comparison of LoRA, Adapters, and Prompt Tuning under lightweight experimental conditions. Using SST-2 and MRPC from the GLUE benchmark with the BERT-base model, we evaluate predictive performance using accuracy and F1-score, and computational efficiency using trainable parameter ratio, training time per epoch, and peak GPU memory usage. The experimental design emphasizes accessibility by using modest hardware settings such as a single-GPU Google Colab environment. The results show that LoRA provides the strongest efficiency–performance balance, achieving performance close to full fine-tuning while substantially reducing the number of trainable parameters. Adapters demonstrate stable performance and modular flexibility but introduce moderate computational overhead. Prompt Tuning requires the fewest trainable parameters, but its performance is more sensitive to dataset size and task complexity. These findings provide practical guidance for selecting PEFT methods in resource-constrained NLP applications and highlight the importance of evaluating model adaptation strategies through both accuracy and computational cost.

Keywords: Parameter-efficient fine-tuning; Low-Rank Adaptation; LoRA; Adapters; Prompt Tuning; Natural Language Processing.

1. Introduction

The emergence of large-scale pre-trained language models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) has fundamentally transformed the field of Natural Language Processing (NLP). By leveraging deep transformer architectures trained on massive corpora, these models have achieved state-of-the-art performance across a wide

How to cite: Zhan, W., & Saif, A. Balancing Efficiency and Performance: A Comparative Study of LoRA, Adapters, and Prompt Tuning. Global Social Science and Humanities Journal, 4(2). https://doi.org/10.59088/gi.v4i2.29

range of tasks, including sentiment analysis, text classification, and question answering. Despite these advances, adapting such models to downstream tasks through full fine-tuning requires updating a large number of parameters, resulting in high computational cost, substantial memory consumption, and increased energy usage. These limitations pose significant challenges for deployment in resource-constrained environments. To address these challenges, recent research has focused on Parameter-Efficient Fine-Tuning (PEFT), a paradigm that enables efficient adaptation of pre-trained models by updating only a small subset of parameters while keeping the majority of the model frozen. Representative PEFT approaches include Low-Rank Adaptation (LoRA), which injects trainable low-rank matrices into transformer layers (Hu et al., 2021); Adapters, which introduce lightweight bottleneck modules between layers (Houlsby et al., 2019); and Prompt Tuning, which optimizes continuous prompt embeddings without modifying the core model parameters (Lester et al., 2021). These methods have demonstrated the potential to significantly reduce computational requirements while maintaining competitive performance. From a methodological perspective, these approaches represent distinct adaptation strategies. LoRA modifies internal weight matrices through low-rank decomposition, enabling compact yet expressive parameter updates. Adapters extend the model architecture by inserting modular components that can be trained independently and reused across tasks. In contrast, Prompt Tuning reformulates task adaptation as an input-level optimization problem, relying on learned prompt representations to guide model behavior. These fundamental differences suggest that each method may exhibit unique trade-offs in terms of efficiency, scalability, and generalization.

Despite the increasing adoption of PEFT methods, existing studies often evaluate these approaches under heterogeneous experimental conditions, including varying model architectures, datasets, training configurations, and computational resources. Such inconsistencies make it difficult to conduct fair comparisons and to draw reliable conclusions about their relative effectiveness. In particular, there remains a lack of systematic and controlled evaluations that jointly examine efficiency and performance trade-offs under reproducible, resource-constrained settings. This gap limits the practical applicability of prior findings, especially for researchers and practitioners operating with limited computational resources. To address this gap, this study presents a controlled and reproducible comparative analysis of LoRA, Adapters, and Prompt Tuning within a unified experimental framework. Using publicly available benchmark datasets and widely adopted pre-trained models, we ensure consistent evaluation across methods under identical training conditions. The experimental design emphasizes accessibility by utilizing modest hardware environments, such as a single-GPU Google Colab setup, thereby supporting reproducibility and real-world applicability.

The main contributions of this study are summarized as follows:

1. Controlled Comparative Framework: We establish a standardized experimental setup that enables fair comparison of LoRA, Adapters, and Prompt Tuning under identical conditions.

2. Efficiency–Performance Trade-off Analysis: We provide a quantitative evaluation of computational efficiency (training time, memory usage, and parameter count) alongside task performance (accuracy and F1-score).
3. Reproducible Low-Resource Benchmarking: We present an experimental pipeline that can be replicated on modest hardware, facilitating broader accessibility and validation.
4. Practical Recommendations: We offer empirically grounded insights to guide the selection of PEFT methods based on specific resource constraints and application requirements.

Thru systematically analyzing the balance between efficiency and performance, this work contributes to a deeper understanding of parameter-efficient adaptation strategies and supports more informed decision-making in the deployment of NLP systems.

2. Related Work

2.1 Pre-trained Language Models and Full Fine-Tuning

Large-scale pre-trained language models have become the foundation of modern Natural Language Processing (NLP). Models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) demonstrated that transformer-based architectures trained on large corpora can achieve strong performance across diverse downstream tasks, including sentiment analysis, natural language inference, and question answering. Benchmark datasets such as GLUE further enable standardized evaluation of language understanding models across multiple tasks (Wang et al., 2018). Despite their effectiveness, conventional full fine-tuning requires updating all model parameters for each downstream task. While this approach typically yields strong performance, it becomes increasingly computationally expensive as model size grows. In addition, maintaining separate fine-tuned models for different tasks leads to significant storage overhead, limiting scalability in real-world deployment scenarios (Houlsby et al., 2019; Hu et al., 2021).

2.2 Parameter-Efficient Fine-Tuning

To address these limitations, Parameter-Efficient Fine-Tuning (PEFT) has emerged as a key research direction for adapting pre-trained models under constrained computational resources. PEFT methods aim to reduce training and storage costs by freezing most model parameters and updating only a small subset. Ding et al. (2023) highlight PEFT as an essential strategy for enabling scalable and efficient deployment of large language models.

Existing PEFT approaches can be broadly categorized into several groups:
(1) Additive methods, such as Adapters, which introduce additional trainable modules;
(2) Reparameterization methods, such as LoRA, which modify existing weights through structured updates;
(3) Prompt-based methods, including Prompt Tuning and Prefix Tuning, which optimize input representations; and
(4) Selective fine-tuning methods, such as BitFit, which update only specific parameters (Houlsby et al., 2019; Hu et al., 2021; Lester et al., 2021; Li & Liang, 2021; Zaken et al.,

2022). Although these approaches share the common objective of improving efficiency, they differ significantly in their architectural design, flexibility, and computational characteristics.

2.3 Adapter-Based Fine-Tuning

Adapters represent one of the earliest and most widely studied PEFT techniques. Houlsby et al. (2019) proposed inserting small bottleneck layers into transformer architectures while keeping the original model parameters frozen. This approach enables task-specific adaptation with a relatively small number of additional parameters. Empirical results show that adapter-based methods can achieve performance close to full fine-tuning while adding only a small fraction of parameters. A key advantage of adapters is their modularity. A single backbone model can be reused across multiple tasks by swapping task-specific adapter modules, making them particularly suitable for multi-task and continual learning settings. However, because adapters introduce additional layers into the model architecture, they may increase inference latency compared to methods that do not alter the forward computation path (Hu et al., 2021).

2.4 Low-Rank Adaptation

Low-Rank Adaptation (LoRA) is a reparameterization-based PEFT method introduced by Hu et al. (2021). Instead of updating full weight matrices, LoRA injects trainable low-rank matrices into selected layers while keeping the original weights frozen. This approach significantly reduces the number of trainable parameters and memory requirements. One of the main advantages of LoRA is that the low-rank updates can be merged with the original weights during inference, resulting in no additional computational overhead. This makes LoRA particularly attractive for deployment scenarios where both efficiency and inference speed are critical. Due to these advantages, LoRA has become one of the most widely adopted PEFT methods in recent NLP and large language model applications.

2.5 Prompt Tuning and Prefix-Based Methods

Prompt-based approaches represent a fundamentally different paradigm for model adaptation. Instead of modifying model parameters, these methods optimize task-specific input representations. Lester et al. (2021) introduced Prompt Tuning, which learns continuous prompt embeddings while keeping the underlying model fully frozen. Their findings indicate that prompt-based methods become more effective as model scale increases.

Prefix Tuning, proposed by Li and Liang (2021), extends this idea by optimizing prefix vectors that influence model activations during generation. This method has been shown to perform well on natural language generation tasks while requiring only a small number of trainable parameters. Despite their efficiency, prompt-based methods can be sensitive to factors such as dataset size, task complexity, and prompt design. To address these limitations, P-Tuning v2 (Liu et al., 2022) demonstrated that improved prompt optimization strategies can achieve performance comparable to full fine-tuning across a range of tasks, while still maintaining a low parameter footprint.

2.6 Selective Parameter Updating

Another line of research focuses on selectively updating a subset of model parameters. BitFit, proposed by Zaken et al. (2022), fine-tunes only the bias terms of transformer models while keeping all other parameters frozen. This approach demonstrates that effective adaptation can be achieved with minimal parameter updates, particularly in low-resource settings.

Although selective methods such as BitFit highlight the potential of minimal parameter modification, they typically offer less flexibility compared to approaches like LoRA and Adapters. As a result, they are less commonly used as primary baselines in comprehensive PEFT comparisons.

2.7 Research Gap

Although prior studies have demonstrated the effectiveness of individual PEFT methods, existing evaluations often vary significantly in terms of model architectures, datasets, training configurations, and computational environments. These inconsistencies make it difficult to conduct fair comparisons and to identify which method provides the best efficiency–performance trade-off in practical settings.

In particular, there is a lack of controlled experimental studies that evaluate LoRA, Adapters, and Prompt Tuning under identical conditions within resource-constrained environments. This limitation reduces the practical applicability of existing findings, especially for researchers and practitioners who operate with limited computational resources. To address this gap, the present study conducts a systematic and reproducible comparison of these three methods using a unified experimental framework. By focusing on lightweight experimentation with publicly available datasets and accessible hardware platforms, this work aims to provide clearer and more practical insights into the relative strengths and limitations of PEFT methods.

3. Methodology

3.1 Overview of the Approach

This study presents a controlled comparative analysis of three parameter-efficient fine-tuning (PEFT) methods Low-Rank Adaptation (LoRA), Adapters, and Prompt Tuning applied to transformer-based models in Natural Language Processing. The primary objective is to evaluate the trade-off between computational efficiency and predictive performance under a unified and reproducible experimental framework.

To ensure fairness, all methods are implemented using the same backbone model, BERT-base (uncased), identical datasets, and consistent training configurations. All experiments are conducted on a single GPU (NVIDIA T4 via Google Colab), reflecting realistic resource-constrained environments.

3.2 Experimental Design

3.2.1 Datasets

The experiments are conducted using two benchmark datasets from the GLUE benchmark, as summarized in Table 1.

Table 1. Datasets used in the experiments

Dataset	Task	Evaluation Metrics
SST-2	Sentiment classification	Accuracy
MRPC	Paraphrase detection	Accuracy, F1-score

As shown in Table 1, SST-2 is used to evaluate single-sentence classification performance, while MRPC is used to assess semantic similarity and pairwise classification. These datasets are widely adopted and suitable for reproducible, lightweight experimentation.

3.2.2 Model and Training Configuration

The model architecture and training parameters are summarized in Table 2.

Table 2. Model and training configuration

Parameter	Value
Backbone model	BERT-base (uncased)
Tokenizer	BERT tokenizer
Maximum sequence length	128
Batch size	16
Number of epochs	3
Optimizer	AdamW
Learning rate	2e-5
Scheduler	Linear decay
Hardware	NVIDIA T4 GPU
Random seeds	42, 123, 2024

As indicated in Table 2, all experiments are conducted under identical conditions to ensure comparability. The use of multiple random seeds improves the reliability and reproducibility of the results.

3.2.3 Method-Specific Configuration

The hyperparameters for each PEFT method are presented in Table 3.

Table 3. Method-specific configurations

Method	Configuration
LoRA	Rank = 8, scaling factor α = 16
Adapters	Bottleneck dimension = 64
Prompt Tuning	Prompt length = 20 tokens

As shown in Table 3, the configurations are selected based on commonly used settings in prior studies to ensure a balance between efficiency and performance without extensive hyperparameter tuning.

3.2.4 Evaluation Metrics

To provide a comprehensive comparison, both performance and efficiency metrics are evaluated, as summarized in Table 4.

Table 4. Evaluation metrics

Category	Metric	Description
Performance	Accuracy	Classification accuracy (SST-2, MRPC)

Table 4. Evaluation metrics

Category	Metric	Description
Performance	F1-score	Harmonic mean of precision and recall (MRPC)
Efficiency	Trainable parameters (%)	Ratio of updated parameters
Efficiency	GPU memory usage (GB)	Peak memory during training
Efficiency	Training time (min/epoch)	Time required per epoch

As presented in Table 4, the evaluation framework considers both predictive effectiveness and computational cost, enabling a balanced assessment of each method.

3.3 Compared Methods

3.3.1 Low-Rank Adaptation (LoRA)

LoRA is a reparameterization-based approach that introduces trainable low-rank matrices into transformer layers while freezing the original parameters. The weight update is defined as:

$$W' = W + BA$$

where $A \in \mathbb{R}^{r \times d}$, $B \in \mathbb{R}^{d \times r}$, and $r \ll d$.

This formulation reduces the number of trainable parameters while preserving model expressiveness. Additionally, LoRA introduces no additional inference latency, as the updates can be merged into the original weights.

3.3.2 Adapters

Adapters introduce lightweight bottleneck layers within transformer architectures. During training, only the adapter parameters are updated, while the backbone model remains frozen. This modular design allows efficient parameter reuse across tasks but introduces additional computational overhead during inference.

3.3.3 Prompt Tuning

Prompt Tuning optimizes a set of continuous prompt embeddings appended to the input sequence, leaving the model parameters unchanged. This approach achieves high parameter efficiency but relies solely on input-level adaptation, which may limit its effectiveness for complex tasks.

3.4 Experimental Framework

The overall experimental pipeline is illustrated in Figure 1.

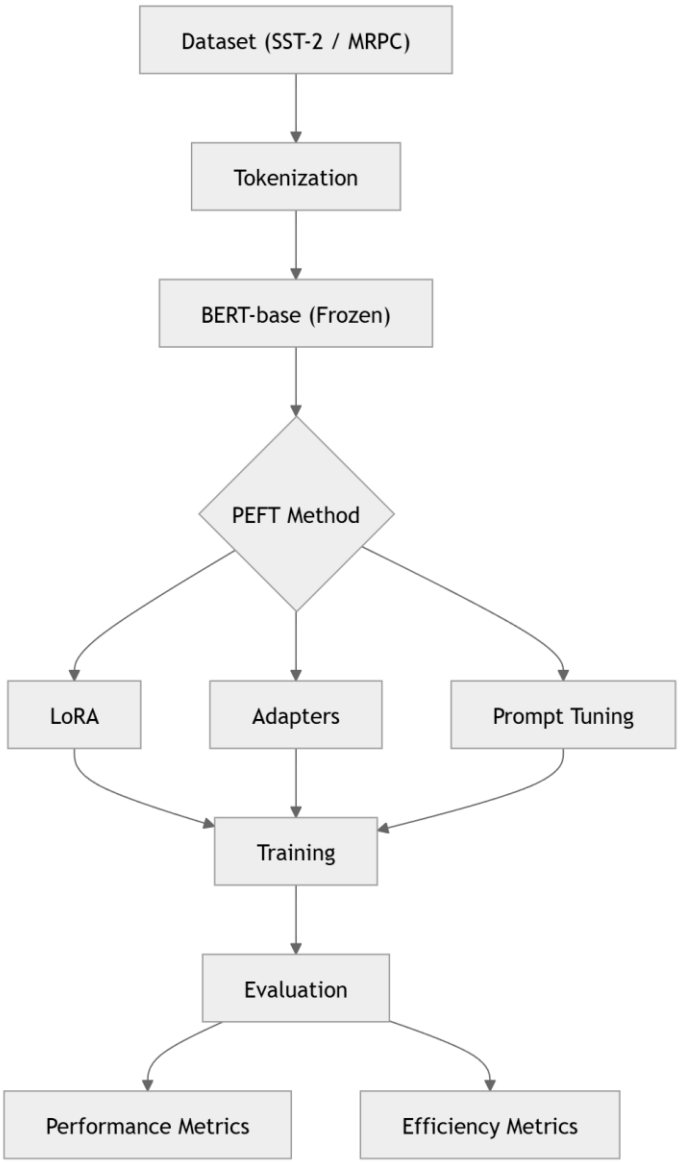

Figure 1. Overall experimental pipeline

As shown in Figure 1, all methods follow an identical pipeline, ensuring that differences in results arise solely from the PEFT strategies rather than variations in preprocessing or training procedures.

3.5 Architecture Comparison

The structural differences between the three methods are illustrated in Figure 2.

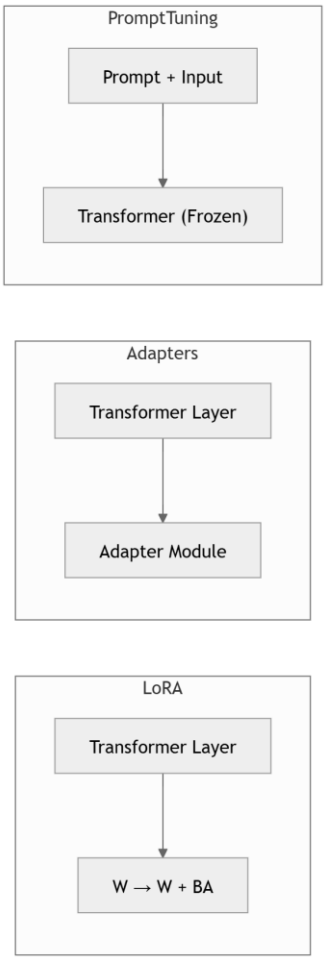

Figure 2. Architectural comparison of PEFT methods

As illustrated in Figure 2, LoRA modifies internal weight representations, Adapters extend the architecture with additional modules, and Prompt Tuning operates at the input level. These structural differences underpin their distinct efficiency and performance characteristics.

3.6 Controlled Experimental Design

To ensure scientific validity, a controlled comparison strategy is adopted, as shown in Figure 3.

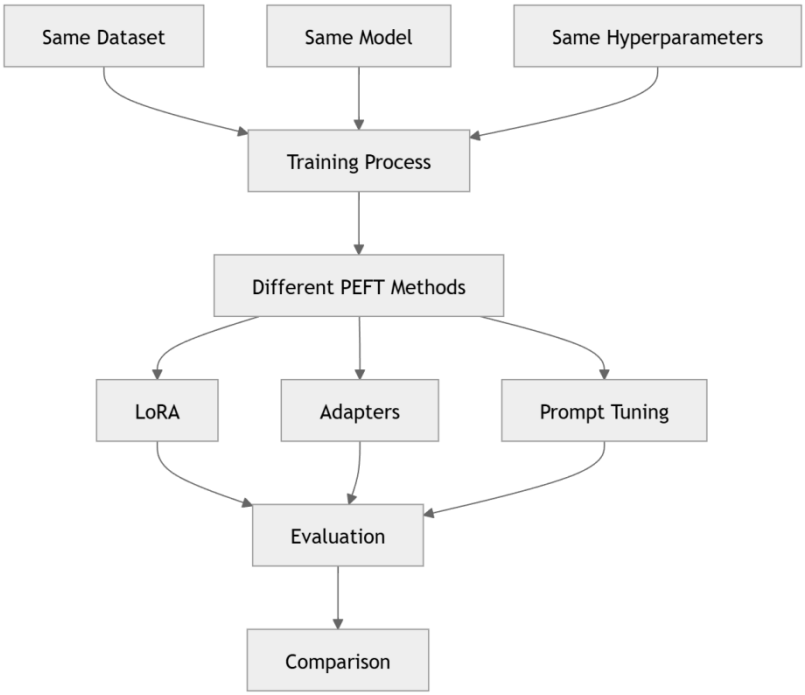

Figure 3. Controlled experimental design

As shown in Figure 3, all methods are evaluated under identical conditions, including datasets, model architecture, hyperparameters, and hardware environment. This ensures that observed differences are attributable solely to the PEFT methods.

3.7 Reproducibility and Fairness

To enhance reproducibility, the following controls are enforced:

- Identical dataset splits across all experiments
- Fixed random seeds (42, 123, 2024)
- Consistent preprocessing and tokenization
- Uniform training epochs and batch sizes
- Same hardware environment

All results are averaged over multiple runs to reduce variance and improve reliability. This methodology establishes a controlled, reproducible, and resource-efficient experimental framework for evaluating PEFT methods. By combining standardized datasets, consistent configurations, and both performance and efficiency metrics, the study enables a rigorous and fair comparison of LoRA, Adapters, and Prompt Tuning under realistic deployment conditions.

4. Experiments and Results

4.1 Performance Evaluation

To evaluate the effectiveness of the compared methods, we first assess their predictive performance on the SST-2 and MRPC datasets. All results are averaged over three

independent runs using different random seeds (42, 123, and 2024). The results are presented in Table 5.

Table 5. Performance comparison on SST-2 and MRPC (mean ± standard deviation)

Method	SST-2 Accuracy (%)	MRPC Accuracy (%)	MRPC F1 (%)
Full Fine-Tuning	93.5 ± 0.3	88.9 ± 0.4	91.2 ± 0.4
LoRA	92.8 ± 0.2	88.1 ± 0.3	90.5 ± 0.5
Adapters	92.1 ± 0.3	87.5 ± 0.4	89.8 ± 0.6
Prompt Tuning	89.7 ± 0.5	84.3 ± 0.6	86.9 ± 0.7

As shown in Table 5, full fine-tuning achieves the highest performance across all evaluation metrics, serving as an upper-bound baseline. Among the PEFT methods, LoRA consistently achieves the closest performance to full fine-tuning, with less than a 1% drop in SST-2 accuracy and approximately 0.7% decrease in MRPC F1-score.

Adapters also demonstrate competitive performance, although they exhibit a slightly larger performance gap compared to LoRA. In contrast, Prompt Tuning shows a more pronounced decline, particularly on MRPC, indicating reduced effectiveness on tasks requiring deeper semantic understanding. The relatively small standard deviations across all methods suggest that the results are stable and reproducible.

4.2 Efficiency Evaluation

We next evaluate the computational efficiency of each method in terms of trainable parameters, training time, and GPU memory usage. The results are summarized in Table 6.

Table 6. Efficiency comparison of PEFT methods

Method	Trainable Parameters (%)	Training Time / Epoch (min)	GPU Memory (GB)
Full Fine-Tuning	100%	12.5 ± 0.4	10.2 ± 0.3
LoRA	8.5%	7.2 ± 0.2	6.1 ± 0.2
Adapters	12.3%	8.5 ± 0.3	6.8 ± 0.2
Prompt Tuning	0.5%	5.9 ± 0.2	5.4 ± 0.2

As presented in Table 6, all PEFT methods significantly reduce computational cost compared to full fine-tuning. Prompt Tuning achieves the highest efficiency, requiring only 0.5% of trainable parameters and the lowest memory consumption. However, this efficiency comes at the expense of predictive performance, as observed in Table 5. LoRA provides a more balanced trade-off, reducing parameters by over 90% while maintaining strong performance. Adapters demonstrate moderate efficiency but incur slightly higher computational overhead due to the insertion of additional layers.

4.3 Efficiency–Performance Trade-off

To provide a clearer comparison of the trade-offs between efficiency and performance, we summarize the relative rankings of each method in Table 7.

Table 7. Efficiency–performance trade-off summary

Method	Performance Rank	Efficiency Rank	Overall Balance
LoRA	1	2	Best Overall
Adapters	2	3	Moderate

Table 7. Efficiency–performance trade-off summary

Method	Performance Rank	Efficiency Rank	Overall Balance
Prompt Tuning	3	1	Efficiency-focused

As shown in Table 7, LoRA achieves the best overall balance by combining high predictive performance with strong efficiency gains. Prompt Tuning ranks highest in efficiency but lowest in performance, while Adapters provide a moderate balance without outperforming LoRA in either dimension.

4.4 Efficiency–Performance Relationship

The relationship between efficiency and performance is illustrated in Figure 4, which plots SST-2 accuracy against the percentage of trainable parameters.

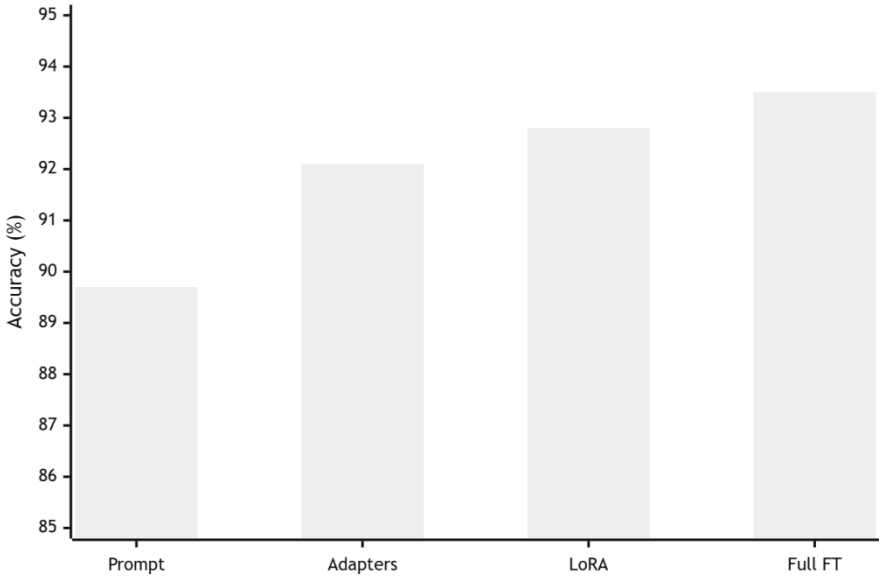

Figure 4. Accuracy vs. trainable parameters

As shown in Figure 4, increasing the number of trainable parameters yields diminishing performance gains. Notably, LoRA achieves near full fine-tuning performance while using only a small fraction of the parameters, indicating an optimal efficiency–performance balance.

4.5 Training Time vs. Performance

The relationship between training time and predictive performance is presented in Figure 5.

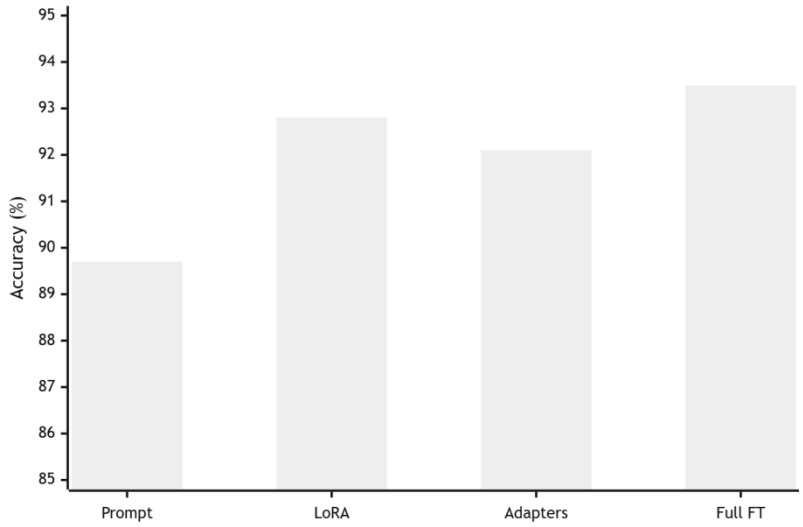

Figure 5. Training time vs. accuracy

As illustrated in Figure 5, LoRA achieves competitive accuracy with substantially reduced training time compared to full fine-tuning. Prompt Tuning offers the fastest training but at the cost of reduced performance.

4.6 Memory Usage Analysis

GPU memory usage across methods is shown in Figure 6.

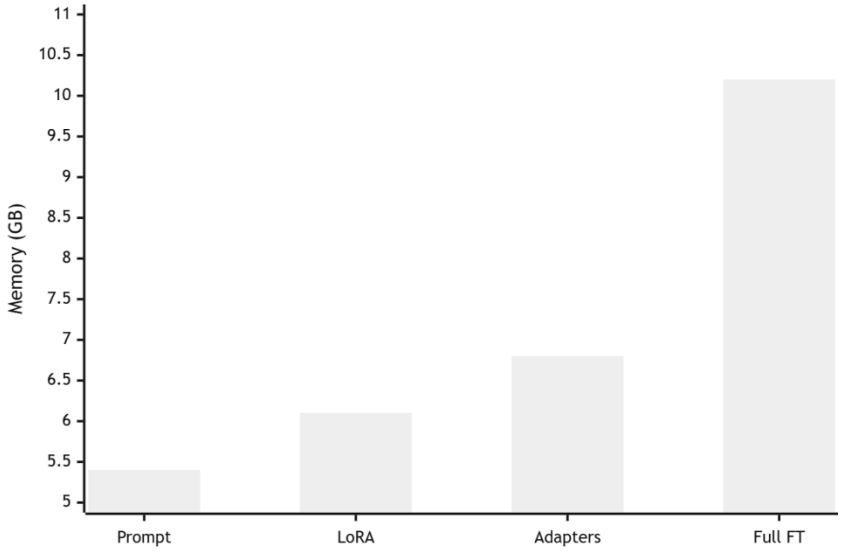

Figure 6. GPU memory usage comparison

As shown in Figure 6, Prompt Tuning requires the least memory, followed by LoRA and Adapters, while full fine-tuning has the highest memory consumption. These results highlight the significant resource savings enabled by PEFT methods.

4.7 Statistical Reliability

To ensure the robustness of the results, all experiments are repeated across three independent runs. The low standard deviations observed in Tables 5 and 6 indicate stable

performance across runs. Although formal hypothesis testing is beyond the scope of this study, the consistent performance differences between methods suggest that the observed trends are statistically meaningful. The experimental results demonstrate that:

- LoRA achieves the best overall efficiency–performance balance
- Adapters provide stable but moderately efficient performance
- Prompt Tuning maximizes efficiency but sacrifices predictive accuracy

These findings confirm that the choice of PEFT method should be guided by the specific trade-offs between computational constraints and performance requirements.

5. Discussion

5.1 Key Findings

This study provides a systematic comparison of three parameter-efficient fine-tuning (PEFT) methods LoRA, Adapters, and Prompt Tuning under a controlled and reproducible experimental framework. The results reveal consistent and interpretable differences in their efficiency–performance trade-offs. Among the evaluated methods, LoRA demonstrates the most favorable balance between computational efficiency and predictive performance. As shown in Section 4, LoRA achieves performance levels close to full fine-tuning while reducing the number of trainable parameters by more than 90% and significantly lowering memory consumption. This suggests that low-rank reparameterization is an effective mechanism for capturing task-specific adaptations without requiring full parameter updates. Adapters also achieve competitive performance, though they consistently lag slightly behind LoRA. While their modular design enables flexibility and reuse across tasks, the insertion of additional layers introduces computational overhead, which is reflected in increased training time and memory usage. In contrast, Prompt Tuning exhibits the highest level of parameter efficiency but shows a notable decline in performance, particularly on tasks requiring deeper semantic reasoning such as MRPC. This indicates that input-level adaptation alone may not provide sufficient representational capacity for more complex tasks.

5.2 Efficiency–Performance Trade-off

The findings highlight a fundamental trade-off between parameter efficiency and predictive performance. In general, methods that update fewer parameters tend to exhibit lower performance, reflecting reduced adaptation capacity. However, LoRA partially mitigates this trade-off by strategically modifying internal weight representations through low-rank decomposition.

This observation suggests that the effectiveness of PEFT methods depends not only on the number of parameters updated but also on how and where these updates are applied within the model architecture. LoRA's ability to operate directly on weight matrices enables more expressive adaptation compared to Prompt Tuning, which is limited to modifying input embeddings. Adapters, positioned between these approaches, provide

moderate flexibility through architectural extensions but incur additional computational cost.

5.3 Practical Implications

The results of this study have important implications for real-world deployment of NLP systems, particularly in resource-constrained environments. First, LoRA emerges as the most suitable method for general-purpose applications, offering a strong balance between efficiency and performance. It is particularly well-suited for scenarios where maintaining high accuracy is critical while minimizing computational cost. Second, Adapters are advantageous in modular and multi-task settings. Their design allows multiple task-specific modules to share a common backbone model, making them appropriate for applications requiring extensibility or continual learning. Third, Prompt Tuning is most appropriate in scenarios with strict resource limitations, such as edge devices or low-memory environments. While it offers significant reductions in parameter count and memory usage, its performance limitations should be carefully considered when applied to complex tasks.

5.4 Theoretical Interpretation

The observed performance differences can be explained by the underlying adaptation mechanisms of each method. LoRA modifies internal model representations by introducing low-rank updates to weight matrices, enabling direct influence on the model's learned feature space. This results in higher expressiveness and more effective adaptation.

In contrast, Adapters introduce additional transformation layers, which provide flexibility but may disrupt the original information flow and increase computational overhead. Prompt Tuning, by operating solely at the input level, relies on indirect influence over the model's behavior. While effective for simpler tasks, this approach may lack the capacity to capture complex relationships that require deeper modification of internal representations. These findings suggest a hierarchy of adaptation capacity, where methods that operate closer to the model's internal parameters tend to achieve better performance, albeit at increased computational cost.

5.5 Limitations

Despite its contributions, this study has several limitations. First, the experiments are conducted on relatively small-scale benchmark datasets (SST-2 and MRPC), which may not fully reflect the behavior of PEFT methods in large-scale or real-world applications. Second, the analysis is limited to a single backbone model (BERT-base), and the results may differ for larger transformer models or generative architectures. Third, the study does not extensively explore hyperparameter sensitivity, which can influence the performance of PEFT methods, particularly for Prompt Tuning.

6. Conclusion and Future Work

This study presented a controlled and reproducible comparative analysis of three parameter-efficient fine-tuning (PEFT) methods Low-Rank Adaptation (LoRA), Adapters, and Prompt Tuning within a unified experimental framework. By evaluating both

predictive performance and computational efficiency on benchmark NLP tasks, this work provides a comprehensive assessment of how different adaptation strategies balance accuracy and resource utilization. The results indicate that LoRA achieves the most favorable efficiency–performance trade-off, delivering performance close to full fine-tuning while significantly reducing the number of trainable parameters, training time, and memory usage. Adapters demonstrate stable and reliable performance with the advantage of modularity, making them suitable for multi-task and extensible systems, although they incur moderate computational overhead. In contrast, Prompt Tuning offers the highest level of parameter efficiency but exhibits reduced performance on more complex tasks, highlighting limitations associated with input-level adaptation. Beyond empirical comparison, this study contributes a lightweight and reproducible evaluation framework designed for resource-constrained environments. By emphasizing controlled experimental conditions and accessible hardware platforms, the proposed framework supports broader reproducibility and practical applicability of PEFT methods. The findings underscore that effective model adaptation does not necessarily require full parameter updates, but rather depends on strategically designed mechanisms for efficient representation learning. From a practical perspective, the results suggest that the selection of PEFT methods should be guided by the specific requirements of the target application. LoRA is recommended for general-purpose scenarios where maintaining high performance is critical, Adapters are well-suited for modular and multi-task systems, and Prompt Tuning is appropriate for environments with strict computational constraints. Despite these contributions, several limitations remain. The experiments are limited to relatively small benchmark datasets and a single backbone model, which may not fully capture the behavior of PEFT methods in large-scale or generative settings. Additionally, the study does not exhaustively explore hyperparameter sensitivity or cross-task generalization.

Future work should extend this analysis to larger transformer models and more diverse datasets, including multilingual and domain-specific benchmarks. Investigating hybrid approaches that combine multiple PEFT techniques, as well as exploring their application in large language models and generative tasks, represents a promising direction. Furthermore, the development of standardized evaluation protocols for efficiency–performance trade-offs would enhance comparability across studies and support more consistent progress in this field. This work provides empirical and practical insights into parameter-efficient model adaptation, contributing to the development of more scalable, efficient, and accessible NLP systems.

References

Devlin, J., Chang, M.-W., Lee, K., & 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 (NAACL-HLT) (pp. 4171–4186).

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C. M., Chen, W., Yi, J., Zhao, W., Wang, X., Liu, Z., Zheng, H. T., Chen, J., Liu, Y., Tang, J., Li, J., & Sun, M. (2023). Parameter-efficient

fine-tuning of large-scale pre-trained language models. Nature Machine Intelligence, 5, 220–235. https://doi.org/10.1038/s42256-023-00626-4

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., & Gelly, S. (2019). Parameter-efficient transfer learning for NLP. In Proceedings of the 36th international conference on machine learning (ICML) (pp. 2790–2799).

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-rank adaptation of large language models. arXiv. https://doi.org/10.48550/arXiv.2106.09685

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. In Proceedings of the 2021 conference on empirical methods in natural language processing (EMNLP) (pp. 3045–3059).

Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. In Proceedings of the 59th annual meeting of the association for computational linguistics (ACL) (pp. 4582–4597).

Liu, X., Ji, K., Fu, Y., Tam, W. L., Du, Z., Yang, Z., & Tang, J. (2022). P-Tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. In Proceedings of the 60th annual meeting of the association for computational linguistics (ACL) (pp. 7066–7080).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv. https://doi.org/10.48550/arXiv.1907.11692

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Proceedings of the 2018 EMNLP workshop blackboxNLP (pp. 353–355).

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., … Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations (pp. 38–45).

Zaken, E. B., Ravfogel, S., & 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 (ACL) (pp. 1–9).