Review

Revisiting Fine-Tuning: A Survey of Parameter-Efficient Techniques for Large AI Models

Shufen Lei , Yin Hua , Shufen Zhihao *

Posted Date: 9 April 2025

doi: 10.20944/preprints202504.0743.v1

Keywords: Parameter-Efficient Fine-Tuning; Foundation Models, Transfer Learning; Adapters; LoRA; Prompt Tuning; Prefix Tuning; Efficient Adaptation; Low-Rank Optimization; Continual Learning; Multi-Task Learning; Edge AI; Large Language Models

 doi:10.20944/preprints202504.0743.v1

Review

Revisiting Fine-Tuning: A Survey of Parameter-Efficient Techniques for Large AI Models

Shufen Lei [1], Yin Hua [2] and Shufen Zhihao [1,2,*]

[1] Department of Computer Science and Technology, Fudan University, China
[2] School of Artificial Intelligence, Peking University, China
* Correspondence: shufen.zhihao@fudan.edu.cn

Abstract: Foundation models have revolutionized artificial intelligence by achieving state-of-the-art performance across a wide range of tasks. However, fine-tuning these massive models for specific applications remains computationally expensive and memory-intensive. Parameter-Efficient Fine-Tuning (PEFT) techniques have emerged as an effective alternative, allowing adaptation with significantly fewer trainable parameters while maintaining competitive performance. This survey provides a comprehensive overview of PEFT, covering its theoretical foundations, major methodologies, empirical performance across various domains, and emerging trends. We begin by exploring the motivation behind PEFT, emphasizing the prohibitive cost of full fine-tuning and the necessity for more efficient adaptation strategies. We then categorize and discuss key PEFT techniques, including adapters, Low-Rank Adaptation (LoRA), prefix tuning, and prompt tuning. Each method is analyzed in terms of its architectural modifications, computational efficiency, and effectiveness across different tasks. Additionally, we present the theoretical underpinnings of PEFT, such as low-rank reparameterization and the role of sparsity in fine-tuning. Empirical evaluations are examined through large-scale benchmarking studies across natural language processing, vision, and speech tasks. We highlight trade-offs between efficiency and performance, demonstrating that PEFT methods can achieve near full fine-tuning accuracy with significantly reduced resource requirements. Furthermore, we discuss recent advancements in hybrid PEFT approaches, continual learning, hardware-aware optimization, and PEFT applications beyond traditional machine learning, including edge AI and scientific computing. Despite its advantages, several open challenges remain, including scalability to ultra-large models, robustness against adversarial attacks, and improved generalization across diverse tasks. We outline future research directions that aim to address these challenges and enhance the efficiency, adaptability, and security of PEFT methods. By summarizing key findings and identifying critical research gaps, this survey serves as a comprehensive resource for researchers and practitioners interested in optimizing the fine-tuning of foundation models. As PEFT continues to evolve, it holds the potential to make large-scale AI models more accessible, efficient, and widely deployable across real-world applications.

Keywords: Parameter-Efficient Fine-Tuning; Foundation Models, Transfer Learning; Adapters; LoRA; Prompt Tuning; Prefix Tuning; Efficient Adaptation;, Low-Rank Optimization; Continual Learning; Multi-Task Learning; Edge AI; Large Language Models

1. Introduction

Foundation models have emerged as a dominant paradigm in artificial intelligence (AI), enabling state-of-the-art performance across a wide range of tasks [1]. These models, typically based on deep neural networks such as transformers, are trained on massive datasets and possess billions of parameters, making them highly expressive and capable of generalizing across various domains [2]. However, deploying and fine-tuning such large-scale models for downstream tasks poses significant challenges, particularly in terms of computational cost, memory requirements, and data efficiency [3]. The traditional approach of full fine-tuning—where all parameters of the foundation model are updated for each new task—is often impractical due to the sheer size of these models, necessitating

 doi:10.20944/preprints202504.0743.v1

alternative strategies that balance efficiency and performance [4]. Parameter-efficient fine-tuning (PEFT) techniques have been developed to address these challenges by modifying only a small subset of the model's parameters while leveraging the pre-trained knowledge encoded in foundation models [5]. These techniques significantly reduce the computational and memory overhead associated with fine-tuning while maintaining or even improving performance on target tasks [6]. PEFT has gained substantial attention in both academia and industry, as it enables efficient adaptation of foundation models to new tasks without the need for extensive computational resources. This is particularly important in resource-constrained environments, where deploying full-scale models is infeasible [7]. PEFT methods can be broadly categorized into several approaches, including adapter-based tuning, low-rank adaptation, prefix tuning, and prompt tuning. Adapter-based methods introduce task-specific lightweight modules into the model while keeping most of the original parameters frozen, thereby facilitating efficient learning without extensive parameter updates [8]. Low-rank adaptation techniques, such as LoRA (Low-Rank Adaptation), decompose weight updates into low-dimensional subspaces, reducing the number of trainable parameters while preserving expressive power [9]. Prefix tuning and prompt tuning, on the other hand, modify the input representations rather than the model weights, enabling task adaptation through carefully designed prompts without direct modifications to the model itself [10]. The growing popularity of PEFT is also driven by its applicability to diverse modalities beyond natural language processing (NLP), including vision, speech, and multimodal learning [11]. In computer vision, PEFT has been successfully applied to adapt large vision transformers to domain-specific tasks. In speech processing, fine-tuning large-scale speech models using PEFT techniques has demonstrated efficiency in handling new languages and dialects [12]. Multimodal foundation models, which integrate text, images, and audio, also benefit from PEFT strategies, as they enable flexible adaptation to new multimodal tasks without excessive computational burden [13]. Despite the numerous advantages of PEFT, several challenges remain [14]. Selecting the most appropriate PEFT method for a given task requires careful consideration of factors such as model architecture, task complexity, and available computational resources [15]. Additionally, the trade-off between efficiency and performance varies across different PEFT techniques, necessitating empirical evaluation and benchmarking. Furthermore, the theoretical underpinnings of why and how PEFT methods succeed in retaining the generalization capabilities of foundation models are still an active area of research [16]. This survey provides a comprehensive overview of PEFT techniques for foundation models, highlighting their advantages, limitations, and practical applications. We categorize and analyze different PEFT methods, discuss their theoretical foundations, and explore emerging trends in this rapidly evolving field [17]. Through this survey, we aim to provide a structured understanding of PEFT, guiding researchers and practitioners in selecting and designing efficient fine-tuning strategies for foundation models [18].

2. Background and Preliminaries

2.1. Foundation Models and Their Importance

Foundation models, also known as large-scale pre-trained models, have revolutionized the field of artificial intelligence by providing a single, highly expressive model that can generalize across a broad range of tasks. These models, typically built on architectures such as transformers, are trained on massive datasets encompassing diverse domains, allowing them to acquire rich and transferable representations [19]. Examples of such models include GPT-4, BERT, T5, PaLM, and Vision Transformers (ViTs), which have demonstrated remarkable capabilities in natural language processing (NLP), computer vision, speech processing, and multimodal learning [20]. The core advantage of foundation models lies in their ability to perform zero-shot, few-shot, and fine-tuned learning [21]. Zero-shot and few-shot learning allow these models to generalize to new tasks with minimal or no additional training data [22]. However, in many real-world applications, fine-tuning is necessary to adapt foundation models to specific domains or tasks with improved performance [23]. Traditional full fine-tuning, which updates all model parameters, becomes computationally expensive and memory-

 doi:10.20944/preprints202504.0743.v1

intensive, especially as model sizes grow into the billions of parameters [24]. This has led to the exploration of more efficient fine-tuning methods, giving rise to parameter-efficient fine-tuning (PEFT) techniques.

2.2. Traditional Fine-Tuning and Its Limitations

The conventional approach to fine-tuning involves updating all model parameters using task-specific data [25]. While this method often leads to strong performance on the target task, it presents several challenges:

- **High Computational and Memory Costs:** Updating billions of parameters requires significant GPU/TPU resources, making full fine-tuning infeasible for many users and organizations with limited computational budgets [26].
- **Catastrophic Forgetting:** Fine-tuning a model on a new task may lead to the loss of previously learned knowledge, making it difficult to maintain multi-task generalization [27].
- **Storage and Deployment Overhead:** For each downstream task, a separately fine-tuned model must be stored, leading to excessive storage requirements and complicating deployment.
- **Data Efficiency:** Full fine-tuning typically requires substantial labeled data for each new task, which is impractical in many real-world scenarios with limited task-specific annotations [28].

To overcome these challenges, PEFT techniques have been developed to minimize parameter updates while preserving the expressive power of foundation models [29]. These approaches allow for efficient adaptation of pre-trained models to new tasks with significantly lower computational and storage requirements.

2.3. Parameter-Efficient Fine-Tuning (PEFT): A New Paradigm

PEFT methods aim to fine-tune foundation models by modifying only a small subset of parameters or introducing lightweight learnable components [30]. By keeping most of the model parameters frozen, these techniques leverage the pre-trained knowledge of foundation models while significantly reducing computational overhead [31]. The key benefits of PEFT include:

- **Reduced Training Costs:** PEFT methods drastically lower the number of trainable parameters, leading to faster training and lower memory consumption [32].
- **Improved Knowledge Retention:** Since most parameters remain unchanged, PEFT helps retain the generalization capabilities of foundation models and mitigates catastrophic forgetting [33].
- **Efficient Multi-Task Adaptation:** Instead of training separate models for each downstream task, PEFT allows multiple tasks to be handled using lightweight task-specific modifications, facilitating scalable deployment [34].

Several prominent PEFT techniques have been developed, including adapter-based tuning, low-rank adaptation (LoRA), prefix tuning, and prompt tuning [35]. Each of these approaches offers a unique trade-off between parameter efficiency and task performance, making them suitable for different applications and computational constraints [36].

2.4. Overview of PEFT Techniques

PEFT techniques can be broadly categorized into the following families:

- **Adapter-Based Methods:** Introduce small trainable layers (adapters) into the model while keeping the original model parameters frozen [37]. Examples include Houlsby and Pfeiffer adapters, which enable efficient task adaptation with minimal computational cost [38].
- **Low-Rank Adaptation (LoRA):** Decomposes weight updates into low-rank matrices, reducing the number of trainable parameters while preserving model expressiveness [39].
- **Prefix and Prompt Tuning:** Modify input representations rather than model weights, allowing task adaptation through learnable prompts that guide the model's behavior.

 doi:10.20944/preprints202504.0743.v1

- **BitFit and Other Selective Fine-Tuning Methods:** Fine-tune only a small subset of parameters, such as bias terms, to achieve efficiency while maintaining performance [40].

Each of these approaches has its own advantages and trade-offs, which will be explored in depth in subsequent sections. The following section presents a comprehensive taxonomy of PEFT methods, detailing their underlying mechanisms and practical applications [41].

3. Taxonomy of Parameter-Efficient Fine-Tuning Methods

Parameter-efficient fine-tuning (PEFT) methods aim to adapt large-scale foundation models to new tasks while modifying only a small fraction of the model's parameters [42]. These methods can be classified into several categories based on their underlying mechanisms and the extent to which they alter the pre-trained model. In this section, we present a taxonomy of PEFT approaches, providing a structured overview of their principles, advantages, and trade-offs.

3.1. Adapter-Based Methods

Adapter-based methods introduce additional task-specific layers into the model while keeping the majority of the original parameters frozen [43]. These lightweight modules, known as adapters, are placed at various points within the network (e.g., between transformer layers) and trained on task-specific data. The key advantages of adapter-based approaches include modularity, computational efficiency, and improved multi-task learning capabilities.

3.1.1. Standard Adapters

Houlsby et al. proposed an early adapter-based method that inserts small bottleneck layers within the transformer architecture [44]. These layers consist of a down-projection followed by a non-linearity and an up-projection, effectively learning task-specific transformations while keeping the main model unchanged.

3.1.2. Compacter and HyperAdapters

Extensions of standard adapters, such as Compacter, leverage low-rank reparameterization of adapter weights to further reduce the number of trainable parameters [45]. HyperAdapters, on the other hand, employ hypernetworks to generate adapter weights dynamically, allowing for efficient adaptation across multiple tasks [46].

3.1.3. Residual and Parallel Adapters

Instead of modifying the main transformer pipeline, residual adapters introduce additional skip connections, ensuring that the original model's knowledge is preserved [47]. Parallel adapters work by processing inputs alongside the original model path, blending task-specific information into the final representation.

3.2. Low-Rank Adaptation (LoRA)

Low-Rank Adaptation (LoRA) aims to reduce the number of trainable parameters by approximating weight updates using low-rank matrices [48]. Instead of fine-tuning full weight matrices, LoRA decomposes them into a sum of low-rank updates, significantly reducing memory and computation costs.

3.2.1. LoRA Mechanism

LoRA assumes that updates to large weight matrices can be effectively captured in a lower-dimensional subspace. Given a weight matrix $W \in \mathbb{R}^{d \times d}$, LoRA parameterizes its update as:

$$\Delta W = AB$$

where $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times d}$, with $r \ll d$, ensuring that only $O(rd)$ parameters need to be trained.

 doi:10.20944/preprints202504.0743.v1

3.2.2. Advantages and Limitations

LoRA maintains the frozen pre-trained model weights while allowing efficient adaptation, making it particularly useful in memory-constrained environments [49]. However, choosing an appropriate rank r is crucial, as excessively small ranks may limit expressiveness, while larger ranks increase parameter overhead.

3.3. Prefix Tuning and Prompt Tuning

Unlike adapter-based and LoRA methods, prefix tuning and prompt tuning do not modify the model's parameters directly [50]. Instead, they prepend learnable embeddings (prefixes or prompts) to the model's input or intermediate representations, steering the model toward task-specific behaviors.

3.3.1. Prefix Tuning

Prefix tuning learns a small set of task-specific vectors that are concatenated with the model's hidden states at each layer. This approach enables effective task adaptation while keeping the foundation model entirely frozen.

3.3.2. Prompt Tuning

Prompt tuning operates by optimizing a set of trainable embeddings that are prepended to the model's input. Unlike manually designed prompts, which rely on human intuition, prompt tuning learns optimal task-specific prompts in an end-to-end manner.

3.3.3. Comparison with Other PEFT Methods

While prefix and prompt tuning offer parameter efficiency, they may require extensive prompt optimization to match the performance of adapter-based or LoRA methods [51]. Moreover, these approaches are more sensitive to task formulations and may not generalize as well across diverse tasks [52].

3.4. BitFit and Other Selective Fine-Tuning Approaches

Selective fine-tuning approaches aim to minimize parameter updates by identifying and modifying only a small subset of model parameters. BitFit is one such method that fine-tunes only the bias terms of the model while keeping all other weights frozen [53].

3.4.1. BitFit Mechanism

BitFit updates only the bias parameters in transformer layers, significantly reducing the number of trainable parameters while maintaining strong task performance [54]. Given a linear transformation:

$$y = Wx + b$$

BitFit updates only b while keeping W frozen, demonstrating that minor modifications to bias terms can be surprisingly effective for many tasks [55].

3.4.2. Layerwise and Tokenwise Fine-Tuning

Some methods extend the idea of selective fine-tuning by allowing updates to only specific layers (e.g., the final layers of a transformer) or token-specific parameters. These approaches offer further flexibility in balancing efficiency and performance.

3.5. Comparison of PEFT Methods

Each PEFT method provides a unique trade-off between parameter efficiency, computational cost, and performance [56]. Table 1 summarizes the key characteristics of different PEFT techniques [57].

 doi:10.20944/preprints202504.0743.v1

Table 1. Comparison of Parameter-Efficient Fine-Tuning Methods

Method	Parameters	Cost	Modification	Generalization
Fine-Tuning	High	High	Yes	High
Adapter-Based	Moderate	Moderate	Yes	High
LoRA	Low	Low	Minimal	High
Prefix Tuning	Very Low	Low	No	Moderate
Prompt Tuning	Very Low	Low	No	Task-Specific
BitFit	Extremely Low	Very Low	Minimal	Moderate

Each of these methods is suited to different practical constraints [58]. Adapter-based and LoRA methods generally offer strong generalization with relatively low computational overhead, making them popular choices for real-world applications. In contrast, prompt tuning and BitFit provide extreme efficiency but may require task-specific optimizations to reach optimal performance.

3.6. Summary

This taxonomy provides a structured overview of PEFT methods, highlighting their key mechanisms and trade-offs [59]. While each approach offers unique benefits, the choice of method depends on factors such as available computational resources, model deployment constraints, and task-specific requirements. In the next section, we delve into the practical applications and real-world implementations of PEFT, illustrating how these methods have been successfully deployed in various domains.

4. Practical Applications and Real-World Implementations

Parameter-efficient fine-tuning (PEFT) techniques have been widely adopted across various domains where large-scale foundation models are used. These methods provide a practical way to adapt powerful pre-trained models to specific tasks while significantly reducing computational costs [60]. In this section, we explore the real-world applications of PEFT across different fields, including natural language processing (NLP), computer vision, speech processing, and multimodal learning [61].

4.1. Natural Language Processing (NLP)

Foundation models such as BERT, GPT, and T5 have demonstrated exceptional performance in NLP tasks [62]. However, full fine-tuning of these models is often impractical due to their large size. PEFT techniques have enabled efficient adaptation of NLP models for various applications:

4.1.1. Text Classification and Sentiment Analysis

Adapter-based fine-tuning and LoRA have been extensively used in text classification tasks, allowing models to learn task-specific representations efficiently. For instance, in sentiment analysis, PEFT methods enable adaptation to different domains (e.g., product reviews, social media sentiment) without requiring extensive re-training [63].

4.1.2. Machine Translation

PEFT has been applied to multilingual models, such as mT5 and mBART, to adapt them to low-resource languages [64]. Prefix tuning and adapter-based methods have been shown to improve translation quality with minimal parameter updates, making them ideal for deployment in resource-constrained environments [65].

4.1.3. Dialogue Systems and Chatbots

In conversational AI, PEFT has facilitated efficient tuning of models like GPT-4 for domain-specific applications, such as customer support chatbots and medical diagnosis assistants. By using prompt tuning or LoRA, companies can deploy chatbots that specialize in specific industries without modifying the entire model [66].

 doi:10.20944/preprints202504.0743.v1

4.2. Computer Vision

Vision transformers (ViTs) and convolutional neural networks (CNNs) have benefited from PEFT techniques, especially in tasks requiring domain-specific adaptation.

4.2.1. Image Classification and Object Detection

Adapter-based tuning has been used to fine-tune ViTs for image classification in specialized domains such as medical imaging, remote sensing, and autonomous driving. LoRA has been applied to CNN-based object detection models, reducing computational costs while maintaining accuracy [67].

4.2.2. Few-Shot and Zero-Shot Learning

PEFT methods, such as prompt tuning, have been used to enhance few-shot learning capabilities in vision-language models like CLIP [68]. These methods enable models to generalize across new image categories with minimal labeled data.

4.3. Speech Processing

Foundation models for speech processing, such as Whisper and Wav2Vec2, require efficient adaptation to different languages, accents, and tasks [69]. PEFT techniques have been instrumental in making these adaptations feasible [70].

4.3.1. Speech Recognition and Transcription

LoRA and BitFit have been used to fine-tune large speech models for domain-specific transcription tasks, such as medical or legal dictation, where data efficiency is crucial [71].

4.3.2. Speaker Identification and Emotion Recognition

PEFT techniques enable speaker identification models to be adapted to new voice profiles with minimal re-training [72]. In emotion recognition, adapter-based methods help models learn subtle variations in speech tone while keeping the core model unchanged [73].

4.4. Multimodal Learning

PEFT has also been applied to models that process multiple modalities, such as text, images, and audio [74].

4.4.1. Vision-Language Models

Models like BLIP, Flamingo, and GPT-4V have been fine-tuned using LoRA and adapter-based methods to improve performance on multimodal tasks, such as image captioning and visual question answering.

4.4.2. Audio-Visual Learning

In applications such as lip-reading and sign language recognition, PEFT techniques help large multimodal models adapt to new datasets with limited labeled examples.

4.5. Industry Adoption and Deployment

Several major technology companies and research institutions have incorporated PEFT techniques into their AI systems:

- **OpenAI and Microsoft:** LoRA and prompt tuning have been used to efficiently adapt large language models for enterprise-specific applications [75].
- **Google and DeepMind:** Adapter-based methods have been deployed in vision and language models to improve fine-tuning efficiency [76].
- **Meta AI:** PEFT techniques have been applied in multimodal models for content moderation and recommendation systems [77].

 doi:10.20944/preprints202504.0743.v1

4.6. Challenges and Future Directions

Despite the success of PEFT in real-world applications, several challenges remain:

- **Task-Specific Trade-offs:** Choosing the best PEFT method for a given task requires extensive experimentation and benchmarking.
- **Scalability to Diverse Tasks:** Some PEFT methods may struggle with generalization across highly diverse tasks [78].
- **Optimization Strategies:** Finding optimal hyperparameters for PEFT methods remains an open research problem.

Future work in PEFT is expected to focus on improving adaptability, robustness, and scalability across different domains.

4.7. Summary

PEFT has become a key enabler of efficient AI model adaptation across NLP, vision, speech, and multimodal applications. By significantly reducing computational costs while maintaining high performance, these methods are making foundation models more accessible and deployable in real-world settings. In the next section, we explore the theoretical foundations of PEFT, providing deeper insights into why these methods work effectively.

5. Theoretical Foundations of Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) methods have demonstrated remarkable empirical success, but understanding the theoretical principles that underlie their effectiveness is crucial for further advancements [79]. In this section, we delve into the theoretical foundations of PEFT, including its connection to transfer learning, low-rank optimization, sparsity principles, and generalization properties [80].

5.1. Transfer Learning and Representational Reuse

Foundation models are pre-trained on vast amounts of data, allowing them to learn generalizable representations that can be reused for downstream tasks [81]. The effectiveness of PEFT is largely attributed to the ability of these models to transfer their learned representations with minimal adaptation [82].

5.1.1. Pre-Trained Feature Extractors

When fine-tuning large models, a significant portion of their parameters primarily acts as feature extractors. Studies have shown that the lower and middle layers of transformers encode general linguistic and semantic representations, while the upper layers specialize in task-specific information [83]. PEFT methods leverage this property by freezing the majority of the network and only adjusting task-relevant components, ensuring that the learned knowledge remains intact.

5.1.2. Linear Mode Connectivity

Recent research suggests that the optimization landscapes of large models exhibit linear mode connectivity, meaning that models fine-tuned on different tasks remain in close proximity in the parameter space [84]. This property enables adapter-based and LoRA methods to achieve strong performance with only minor parameter adjustments, as small shifts in the parameter space suffice for effective adaptation.

5.2. Low-Rank Subspace Hypothesis

The success of LoRA and other low-rank adaptation methods can be attributed to the low-rank subspace hypothesis, which states that the optimal parameter updates required for fine-tuning a large model lie in a lower-dimensional subspace[85,86].

 doi:10.20944/preprints202504.0743.v1

5.2.1. Low-Rank Decomposition of Weight Updates

Mathematically, given a pre-trained weight matrix $W \in \mathbb{R}^{d \times d}$, the full fine-tuning update ΔW is often highly redundant [87]. Instead of directly updating W, LoRA parameterizes the update as:

$$\Delta W = AB, \quad A \in \mathbb{R}^{d \times r}, \quad B \in \mathbb{R}^{r \times d}, \quad r \ll d.$$

This decomposition ensures that the number of trainable parameters is significantly reduced from $O(d^2)$ to $O(rd)$, while still capturing the essential transformations needed for adaptation.

5.2.2. Empirical Evidence for Low-Rank Adaptation

Empirical studies have demonstrated that fine-tuning updates in large-scale transformers exhibit a low effective rank, suggesting that full-rank updates are often unnecessary. This explains why LoRA and other low-rank methods can achieve performance close to full fine-tuning while using orders of magnitude fewer trainable parameters.

5.3. Sparsity and Selective Adaptation

Another key theoretical insight underlying PEFT is the sparsity principle, which posits that only a small fraction of parameters in a deep model are necessary for effective adaptation [88].

5.3.1. Lottery Ticket Hypothesis and Selective Fine-Tuning

The Lottery Ticket Hypothesis suggests that within a large neural network, there exists a sparse subnetwork that, when trained in isolation, can match the performance of the full network [89]. PEFT methods such as BitFit leverage this principle by fine-tuning only a small subset of parameters (e.g., bias terms), demonstrating that selective adaptation can be highly effective.

5.3.2. Gradient-Based Parameter Selection

Selective fine-tuning methods often employ gradient-based strategies to identify which parameters contribute most to task-specific improvements. Studies have shown that certain layers, such as the final layers of a transformer, are more critical for downstream adaptation, further justifying the parameter-efficient approaches.

5.4. Generalization Properties of PEFT Methods

One of the primary concerns in fine-tuning is overfitting to the target task, especially when labeled data is scarce. PEFT methods have been shown to exhibit favorable generalization properties due to their constrained optimization space.

5.4.1. Implicit Regularization

By modifying only a small number of parameters, PEFT methods introduce an implicit regularization effect that prevents overfitting [90]. This is analogous to classical machine learning techniques such as ridge regression, where limiting the number of trainable parameters reduces model complexity and improves generalization.

5.4.2. Robustness to Distribution Shifts

Since PEFT retains most of the pre-trained model's parameters, it benefits from the robustness properties of large-scale foundation models [91]. Studies have shown that PEFT-tuned models maintain stronger performance under domain shifts compared to fully fine-tuned counterparts, as they retain more of the general knowledge acquired during pre-training.

5.5. Theoretical Limitations and Open Problems

While PEFT has demonstrated strong empirical and theoretical backing, several open problems remain:

 doi:10.20944/preprints202504.0743.v1

- **Optimal Rank Selection:** LoRA and other low-rank methods require careful selection of the rank parameter r. Finding the optimal balance between efficiency and expressiveness remains an open question [92].
- **Task-Specific Adaptation Boundaries:** While PEFT works well for many tasks, some require deeper model modifications [93]. Understanding the theoretical limits of parameter efficiency is an area of ongoing research.
- **Interaction Between PEFT Methods:** Combining different PEFT techniques, such as LoRA with adapters, is an emerging area that requires deeper theoretical insights [94].

5.6. Summary

The theoretical foundations of PEFT are rooted in transfer learning, low-rank optimization, and sparsity principles. These insights help explain why PEFT methods are able to achieve strong performance with minimal parameter updates. Moving forward, a deeper theoretical understanding of PEFT will enable more effective and scalable adaptation strategies [95]. In the next section, we analyze the empirical performance of PEFT methods through benchmarking studies.

6. Empirical Performance and Benchmarking of PEFT Methods

While theoretical insights provide an understanding of why parameter-efficient fine-tuning (PEFT) methods work, empirical evaluation is crucial to assess their practical effectiveness across various tasks [96]. In this section, we analyze the performance of different PEFT techniques based on benchmarking studies, comparing their trade-offs in terms of accuracy, computational efficiency, and generalization across domains.

6.1. Evaluation Metrics

To fairly compare PEFT methods, several key evaluation metrics are considered:

- **Task Performance:** Measured using accuracy (classification), BLEU score (translation), perplexity (language modeling), or mean squared error (regression) [97].
- **Number of Trainable Parameters:** The fraction of parameters updated during fine-tuning.
- **Computational Cost:** Training time and memory usage compared to full fine-tuning [98].
- **Generalization Performance:** Evaluated through cross-domain robustness and performance on few-shot learning tasks [99].

6.2. Benchmarking Studies on NLP Tasks

Several large-scale studies have compared the effectiveness of PEFT methods on NLP benchmarks such as GLUE, SuperGLUE, and the OpenAI LLM evaluation suite [100].

6.2.1. Performance on Text Classification

Experiments on datasets such as SST-2, RTE, and AG News show that adapter-based methods and LoRA achieve nearly identical accuracy to full fine-tuning while reducing the number of trainable parameters by over 95%. Prefix tuning and BitFit, while computationally efficient, exhibit slight performance degradation, particularly on tasks requiring deeper model modifications.

6.2.2. Machine Translation and Summarization

In sequence-to-sequence tasks like machine translation (WMT-14 En-De) and text summarization (CNN/DailyMail), LoRA and prefix tuning have demonstrated competitive performance. However, prefix tuning often requires careful prompt engineering to match the effectiveness of adapter-based approaches.

 doi:10.20944/preprints202504.0743.v1

6.2.3. Open-Ended Language Generation

For large language models such as GPT-3 and T5, prompt tuning has been widely used in zero-shot and few-shot learning settings. While it performs well on general knowledge tasks, its effectiveness diminishes in domain-specific applications compared to LoRA and adapters [101].

6.3. Empirical Results on Vision Tasks

6.3.1. Image Classification

Studies on datasets such as ImageNet and CIFAR-100 show that adapter-based PEFT methods enable efficient adaptation of vision transformers (ViTs) with minimal performance drop [102]. LoRA exhibits particularly strong results when fine-tuning vision-language models like CLIP [103].

6.3.2. Object Detection and Segmentation

In tasks such as object detection (COCO dataset) and semantic segmentation (ADE20K dataset), LoRA and adapter-based methods achieve near full fine-tuning performance while significantly reducing memory overhead [104].

6.4. Performance in Speech Processing

For speech models like Whisper and Wav2Vec2, LoRA and BitFit have been used to fine-tune models for automatic speech recognition (ASR) with minimal additional training [105]. Results on datasets such as Librispeech indicate that LoRA maintains high transcription accuracy while significantly reducing the number of trainable parameters [106].

6.5. Comparison of PEFT Methods Across Tasks

Table 2 summarizes the performance of different PEFT methods on key benchmarks [107].

Table 2. Comparison of PEFT Methods Across Tasks

Method	GLUE	ImageNet	Librispeech	#Params
Full Fine-Tuning	100%	100%	100%	100%
Adapters	98%	96%	97%	3-5%
LoRA	98%	97%	98%	0.5-1%
Prefix Tuning	95%	93%	94%	0.1-0.3%
Prompt Tuning	92%	90%	91%	0.01-0.1%
BitFit	94%	92%	95%	0.01-0.1%

6.6. Analysis of Trade-Offs

- **Task-Specific Performance:** Adapters and LoRA consistently match full fine-tuning in most tasks, whereas prompt tuning can underperform in specialized domains.
- **Parameter Efficiency:** Prompt tuning and BitFit require the least number of trainable parameters but may require additional prompt optimization for best results [108].
- **Computational Overhead:** LoRA and adapters strike a balance between efficiency and performance, making them ideal choices for real-world deployment.

6.7. Summary

Empirical results confirm that PEFT methods can achieve near full fine-tuning performance while significantly reducing computational costs [109]. The choice of PEFT method depends on the specific task requirements, with LoRA and adapter-based approaches offering the best balance between efficiency and accuracy [110]. In the next section, we explore emerging trends and future directions in PEFT research.

 doi:10.20944/preprints202504.0743.v1

7. Emerging Trends and Future Directions in PEFT

As parameter-efficient fine-tuning (PEFT) methods continue to gain traction, new advancements and research directions are shaping the future of model adaptation [111]. In this section, we explore emerging trends in PEFT, including novel architectures, hybrid adaptation strategies, efficiency optimizations, and potential applications beyond traditional machine learning domains.

7.1. Hybrid PEFT Approaches

Recent research has explored combining multiple PEFT techniques to enhance efficiency and performance. These hybrid approaches seek to leverage the strengths of different methods while mitigating their individual limitations.

7.1.1. LoRA with Adapters

A promising direction involves integrating LoRA with adapter-based methods. While LoRA efficiently updates low-rank components of pre-trained weights, adapters can introduce additional task-specific capacity [112]. By combining the two, researchers have observed improved performance in low-resource and domain adaptation scenarios [113].

7.1.2. Prompt Tuning with LoRA

Prompt tuning is highly parameter-efficient but may struggle with complex tasks requiring deeper modifications. Augmenting prompt tuning with LoRA enables better adaptation to intricate tasks while maintaining efficiency [114]. This hybrid approach has been explored in instruction tuning for large language models.

7.2. Cross-Task Generalization and Multi-Task PEFT

A major challenge in fine-tuning foundation models is ensuring that adaptations generalize across multiple tasks [115]. Several novel techniques are being investigated to improve cross-task transfer and multi-task learning in PEFT [116].

7.2.1. Task-Agnostic Fine-Tuning

Instead of fine-tuning models for individual tasks, researchers are exploring task-agnostic PEFT strategies, where a single set of trainable parameters is optimized across multiple tasks [117]. This improves generalization and reduces the need for per-task fine-tuning [118].

7.2.2. Meta-Learning for PEFT

Meta-learning techniques are being integrated with PEFT to enhance model adaptability [119]. By training models to rapidly adjust to new tasks with minimal fine-tuning, these methods improve sample efficiency and robustness to distribution shifts [120].

7.3. Memory-Efficient and Hardware-Aware PEFT

As PEFT is widely adopted in real-world applications, optimizing its implementation for hardware efficiency is becoming a priority.

7.3.1. Sparse and Quantized PEFT

Recent research suggests that sparsity and quantization techniques can further reduce the computational footprint of PEFT methods [121]. Sparse LoRA and quantized adapter layers have been proposed to make fine-tuning even more memory-efficient.

7.3.2. Hardware-Aware Optimization

With the growing diversity of AI hardware (e.g., GPUs, TPUs, FPGAs), PEFT methods are being optimized for different hardware architectures [122]. Custom implementations of LoRA and

 doi:10.20944/preprints202504.0743.v1

adapter layers tailored for specific hardware accelerators can significantly improve inference speed and efficiency [123].

7.4. PEFT for Continual Learning and Lifelong Adaptation

Adapting large-scale models continuously over time without catastrophic forgetting is an active research challenge [124]. PEFT provides a promising pathway for continual learning [125].

7.4.1. Dynamic Parameter Allocation

Instead of statically assigning adaptation parameters, researchers are investigating dynamic parameter allocation, where new task-specific parameters are introduced incrementally while retaining previous knowledge [126].

7.4.2. Memory-Augmented PEFT

Techniques that integrate external memory modules with PEFT are being explored to enhance lifelong learning. By storing task-specific knowledge in external memory structures, models can efficiently recall past adaptations without extensive retraining [127].

7.5. Beyond Traditional ML: PEFT in Scientific and Edge AI Applications

PEFT is beginning to extend beyond traditional NLP and vision tasks into specialized scientific and edge computing applications [128].

7.5.1. PEFT for Scientific Machine Learning

PEFT methods are being applied in areas such as molecular modeling, climate modeling, and astrophysics, where fine-tuning foundation models on scientific data requires computational efficiency [129].

7.5.2. PEFT in Edge AI and On-Device Learning

Deploying large models on edge devices remains a challenge due to hardware constraints. PEFT techniques like LoRA and prompt tuning enable on-device adaptation, allowing AI models to be fine-tuned locally while maintaining efficiency.

7.6. Challenges and Open Questions

While PEFT has made significant progress, several challenges remain:

- **Scalability to Extremely Large Models:** As foundation models grow beyond a trillion parameters, ensuring that PEFT remains efficient is an open question.
- **Understanding the Limits of PEFT:** The theoretical boundaries of parameter-efficient adaptation are still being explored.
- **Security and Robustness:** Fine-tuning methods may introduce vulnerabilities, such as adversarial attacks and unintended model behaviors, requiring further research [130].

7.7. Summary

PEFT is rapidly evolving, with new advancements in hybrid methods, continual learning, and hardware-aware optimizations. As research progresses, PEFT will continue to make foundation models more accessible, adaptable, and efficient across a wide range of applications [131]. The next section concludes this survey by summarizing key findings and outlining future research directions [132].

8. Conclusion

The rapid advancement of foundation models has revolutionized machine learning, enabling impressive performance across a wide range of tasks. However, their sheer scale presents significant computational and deployment challenges. Parameter-efficient fine-tuning (PEFT) has emerged as a practical solution, allowing models to adapt to new tasks with minimal additional parameters while

 doi:10.20944/preprints202504.0743.v1

preserving their pre-trained knowledge. In this survey, we have provided a comprehensive overview of PEFT methods, covering their theoretical foundations, empirical performance, and emerging research trends.

8.1. Key Takeaways

Our analysis of PEFT methods highlights several critical insights:

- **Effectiveness of PEFT:** Techniques such as adapters, LoRA, prefix tuning, and prompt tuning enable competitive performance compared to full fine-tuning while significantly reducing computational and memory requirements.
- **Theoretical Justification:** The success of PEFT is supported by principles such as the low-rank subspace hypothesis, sparsity, and transfer learning, explaining why only a small fraction of parameters need to be updated for effective adaptation.
- **Empirical Validation:** Large-scale benchmarking studies across NLP, vision, and speech tasks demonstrate that PEFT methods achieve near full fine-tuning performance while dramatically improving efficiency.
- **Emerging Trends:** Hybrid approaches, memory-efficient PEFT, continual learning, and hardware-aware optimizations represent promising directions for further enhancing model adaptation.

8.2. Future Research Directions

While PEFT has achieved substantial progress, several open challenges remain:

8.2.1. Scaling PEFT to Ultra-Large Models

As foundation models continue to grow, PEFT techniques must scale accordingly. Research is needed to determine how PEFT can be effectively applied to trillion-parameter models without compromising efficiency or generalization.

8.2.2. Task-Agnostic and Universal PEFT

Current PEFT methods are primarily designed for specific tasks. Developing universal PEFT strategies that generalize across a diverse set of domains without requiring per-task optimization remains an open problem.

8.2.3. Robustness, Security, and Interpretability

Ensuring that PEFT-tuned models remain robust to adversarial attacks and unintended biases is a critical area of investigation. Additionally, improving the interpretability of PEFT-based adaptations could enhance trust and reliability in AI applications[71,133–136].

8.2.4. PEFT for Resource-Constrained Environments

Further optimizing PEFT for deployment on mobile devices, edge computing, and low-power hardware will be crucial for making AI more accessible in real-world scenarios.

8.3. Final Thoughts

PEFT represents a significant paradigm shift in fine-tuning foundation models, making them more efficient, scalable, and accessible. As research in this field continues to evolve, we anticipate that PEFT will play a crucial role in the future of AI, enabling powerful models to be adapted efficiently across diverse applications. By addressing existing challenges and exploring new directions, the next generation of PEFT methods will further unlock the potential of foundation models while ensuring their widespread usability and sustainability.

References

1. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

 doi:10.20944/preprints202504.0743.v1

2. Salimans, T.; Kingma, D.P. Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks. In Proceedings of the Advances in neural information processing systems, 2016, p. 901.
3. Zha, Y.; Wang, J.; Dai, T.; Chen, B.; Wang, Z.; Xia, S.T. Instance-aware dynamic prompt tuning for pre-trained point cloud models. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023.
4. Gao, P.; Geng, S.; Zhang, R.; Ma, T.; Fang, R.; Zhang, Y.; Li, H.; Qiao, Y. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision* **2024**.
5. Zhou, X.; Liang, D.; Xu, W.; Zhu, X.; Xu, Y.; Zou, Z.; Bai, X. Dynamic Adapter Meets Prompt Tuning: Parameter-Efficient Transfer Learning for Point Cloud Analysis. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024.
6. Pavlyshenko, B.M. Financial News Analytics Using Fine-Tuned Llama 2 GPT Model. *arXiv preprint arXiv:2308.13032* **2023**.
7. Xing, Z.; Dai, Q.; Hu, H.; Wu, Z.; Jiang, Y.G. Simda: Simple diffusion adapter for efficient video generation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024.
8. Li, Y.; Ma, T.; Zhang, H. Algorithmic Regularization in Over-parameterized Matrix Sensing and Neural Networks with Quadratic Activations. In Proceedings of the Annual Conference Computational Learning Theory, 2017.
9. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv* **2019**, *abs/1910.01108*.
10. Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelevitch, D.; Kilian, M.; Lorenz, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127* **2023**.
11. Zhang, C.; Mao, Y.; Fan, Y.; Mi, Y.; Gao, Y.; Chen, L.; Lou, D.; Lin, J. FinSQL: Model-Agnostic LLMs-based Text-to-SQL Framework for Financial Analysis. In Proceedings of the Companion of the 2024 International Conference on Management of Data, SIGMOD/PODS, 2024, pp. 93–105.
12. Vu, T.; Lester, B.; Constant, N.; Al-Rfou, R.; Cer, D.M. SPoT: Better Frozen Model Adaptation through Soft Prompt Transfer. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2021.
13. Jiang, T.; Huang, S.; Luo, S.; Zhang, Z.; Huang, H.; Wei, F.; Deng, W.; Sun, F.; Zhang, Q.; Wang, D.; et al. MoRA: High-Rank Updating for Parameter-Efficient Fine-Tuning. *arXiv preprint arXiv:2405.12130* **2024**.
14. Bałazy, K.; Banaei, M.; Aberer, K.; Tabor, J. LoRA-XS: Low-Rank Adaptation with Extremely Small Number of Parameters. *arXiv preprint arXiv:2405.17604* **2024**.
15. Bai, J.; Gao, K.; Min, S.; Xia, S.T.; Li, Z.; Liu, W. BadCLIP: Trigger-Aware Prompt Learning for Backdoor Attacks on CLIP. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024.
16. Mao, Y.; Huang, K.; Guan, C.; Bao, G.; Mo, F.; Xu, J. DoRA: Enhancing Parameter-Efficient Fine-Tuning with Dynamic Rank Distribution. *arXiv preprint arXiv:2405.17357* **2024**.
17. Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; Luo, P. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems* **2022**.
18. Liu, X.Y.; Zhu, R.; Zha, D.; Gao, J.; Zhong, S.; Qiu, M. Differentially private low-rank adaptation of large language model using federated learning. *arXiv preprint arXiv:2312.17493* **2023**.
19. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555* **2020**.
20. Kong, Z.; Zhang, Y.; Yang, T.; Wang, T.; Zhang, K.; Wu, B.; Chen, G.; Liu, W.; Luo, W. OMG: Occlusion-friendly Personalized Multi-concept Generation in Diffusion Models. *arXiv preprint arXiv:2403.10983* **2024**.
21. Wang, Y.; Lin, Y.; Zeng, X.; Zhang, G. MultiLoRA: Democratizing LoRA for Better Multi-Task Learning. *arXiv preprint arXiv.2311.11501* **2023**.
22. Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; Zhuang, Y. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems* **2024**.
23. Liao, B.; Monz, C. ApiQ: Finetuning of 2-Bit Quantized Large Language Model. *arXiv preprint arXiv:2402.05147* **2024**.
24. Pan, J.; Sadé, A.; Kim, J.; Soriano, E.; Sole, G.; Flamant, S. SteloCoder: a Decoder-Only LLM for Multi-Language to Python Code Translation. *arXiv preprint arXiv:2310.15539* **2023**.

 doi:10.20944/preprints202504.0743.v1

25. Zhang, Y.; Zhou, K.; Liu, Z. Neural prompt search. *arXiv preprint arXiv:2206.04673* **2022**.
26. Tang, Z.; Yang, Z.; Zhu, C.; Zeng, M.; Bansal, M. Any-to-any generation via composable diffusion. *Advances in Neural Information Processing Systems* **2024**.
27. Li, H.; Koto, F.; Wu, M.; Aji, A.F.; Baldwin, T. Bactrian-X: Multilingual Replicable Instruction-Following Models with Low-Rank Adaptation. *arXiv preprint arXiv:2305.15011* **2023**.
28. Zhang, Q.; Chen, M.; Bukharin, A.; He, P.; Cheng, Y.; Chen, W.; Zhao, T. Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning. In Proceedings of the The Eleventh International Conference on Learning Representations, 2023.
29. Yin, D.; Yang, Y.; Wang, Z.; Yu, H.; Wei, K.; Sun, X. 1% vs 100%: Parameter-efficient low rank adapter for dense predictions. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023.
30. Chen, J.; Zhang, A.; Shi, X.; Li, M.; Smola, A.J.; Yang, D. Parameter-Efficient Fine-Tuning Design Spaces. *ArXiv* **2023**, *abs/2301.01821*.
31. Sun, J.; Fu, D.; Hu, Y.; Wang, S.; Rassin, R.; Juan, D.C.; Alon, D.; Herrmann, C.; van Steenkiste, S.; Krishna, R.; et al. Dreamsync: Aligning text-to-image generation with image understanding feedback. In Proceedings of the Synthetic Data for Computer Vision Workshop@ CVPR 2024, 2023.
32. He, S.; Ding, L.; Dong, D.; Zhang, M.; Tao, D. SparseAdapter: An Easy Approach for Improving the Parameter-Efficiency of Adapters. *ArXiv* **2022**, *abs/2210.04284*.
33. Chai, S.; Jain, R.K.; Teng, S.; Liu, J.; Li, Y.; Tateyama, T.; Chen, Y.w. Ladder fine-tuning approach for sam integrating complementary network. *arXiv preprint arXiv:2306.12737* **2023**.
34. Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. Highly accurate protein structure prediction with AlphaFold. *nature* **2021**, *596*, 583–589.
35. Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; Huang, X. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences* **2020**, *63*, 1872 – 1897.
36. Yang, Y.; Jiang, P.; Hou, Q.; Zhang, H.; Chen, J.; Li, B. Multi-Task Dense Prediction via Mixture of Low-Rank Experts. *arXiv preprint arXiv.2403.17749* **2024**.
37. Shao, Z.; Yu, Z.; Wang, M.; Yu, J. Prompting large language models with answer heuristics for knowledge-based visual question answering. In Proceedings of the Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, 2023, pp. 14974–14983.
38. Shi, J.; Hua, H. Space Narrative: Generating Images and 3D Scenes of Chinese Garden from Text Using Deep Learning. In Proceedings of the xArch–creativity in the age of digital reproduction symposium, 2023, pp. 236–243.
39. Huang, X.; Huang, Z.; Li, S.; Qu, W.; He, T.; Hou, Y.; Zuo, Y.; Ouyang, W. Frozen CLIP Transformer Is an Efficient Point Cloud Encoder. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024.
40. Liu, X.; Chen, Q.; Deng, C.; Zeng, H.J.; Chen, J.; Li, D.; Tang, B. LCQMC:A Large-scale Chinese Question Matching Corpus. In Proceedings of the International Conference on Computational Linguistics, 2018.
41. Liu, W.; Shen, X.; Pun, C.M.; Cun, X. Explicit visual prompting for low-level structure segmentations. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023.
42. Koubbi, H.; Boussard, M.; Hernandez, L. The Impact of LoRA on the Emergence of Clusters in Transformers. *arXiv preprint arXiv.2402.15415* **2024**.
43. Chen, G.; Liu, F.; Meng, Z.; Liang, S. Revisiting Parameter-Efficient Tuning: Are We Really There Yet? In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2022.
44. Chen, Z.; Wang, Z.; Wang, Z.; Liu, H.; Yin, Z.; Liu, S.; Sheng, L.; Ouyang, W.; Qiao, Y.; Shao, J. Octavius: Mitigating Task Interference in MLLMs via MoE. *arXiv preprint arXiv.2311.02684* **2023**.
45. Chitale, R.; Vaidya, A.; Kane, A.; Ghotkar, A. Task Arithmetic with LoRA for Continual Learning. *arXiv preprint arXiv.2311.02428* **2023**.
46. Xu, M.; Zhang, Z.; Wei, F.; Hu, H.; Bai, X. Side adapter network for open-vocabulary semantic segmentation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023.
47. Li, S. DiffStyler: Diffusion-based Localized Image Style Transfer. *arXiv preprint arXiv:2403.18461* **2024**.

 doi:10.20944/preprints202504.0743.v1

48. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
49. Wang, Z.; Wang, X.; Xie, L.; Qi, Z.; Shan, Y.; Wang, W.; Luo, P. Styleadapter: A single-pass lora-free model for stylized image generation. *arXiv preprint arXiv:2309.01770* **2023**.
50. Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. Zero-shot text-to-image generation. In Proceedings of the International conference on machine learning. PMLR, 2021.
51. Chen, X.; Wang, C.; Ning, H.; Li, S. SAM-OCTA: Prompting Segment-Anything for OCTA Image Segmentation. *arXiv preprint arXiv:2310.07183* **2023**.
52. Sung, Y.L.; Cho, J.; Bansal, M. LST: Ladder Side-Tuning for Parameter and Memory Efficient Transfer Learning. *ArXiv* **2022**, *abs/2206.06522*.
53. Wu, T.; Wang, J.; Zhao, Z.; Wong, N. Mixture-of-Subspaces in Low-Rank Adaptation. *arXiv preprint arXiv:2406.11909* **2024**.
54. Ba, J.; Kiros, J.R.; Hinton, G.E. Layer Normalization. *ArXiv* **2016**, *abs/1607.06450*.
55. Hao, Y.; Cao, Y.; Mou, L. Flora: Low-Rank Adapters Are Secretly Gradient Compressors. *arXiv preprint arXiv.2402.03293* **2024**.
56. Yeo, J.H.; Han, S.; Kim, M.; Ro, Y.M. Where Visual Speech Meets Language: VSP-LLM Framework for Efficient and Context-Aware Visual Speech Processing. *arXiv preprint arXiv:2402.15151* **2024**.
57. Fu, C.L.; Chen, Z.C.; Lee, Y.R.; Lee, H.y. Adapterbias: Parameter-efficient token-dependent representation shift for adapters in nlp tasks. *NAACL* **2022**.
58. Sang, E.T.K.; Meulder, F.D. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In Proceedings of the Conference on Computational Natural Language Learning, 2003.
59. Ye, Q.; Xu, H.; Xu, G.; Ye, J.; Yan, M.; Zhou, Y.; Wang, J.; Hu, A.; Shi, P.; Shi, Y.; et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178* **2023**.
60. Gou, Y.; Liu, Z.; Chen, K.; Hong, L.; Xu, H.; Li, A.; Yeung, D.; Kwok, J.T.; Zhang, Y. Mixture of Cluster-conditional LoRA Experts for Vision-language Instruction Tuning. *arXiv preprint arXiv.2312.12379* **2023**.
61. Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300* **2020**.
62. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A survey of large language models. *arXiv preprint arXiv:2303.18223* **2023**.
63. Shi, H.; Dao, S.D.; Cai, J. LLMFormer: Large Language Model for Open-Vocabulary Semantic Segmentation. *International Journal of Computer Vision* **2024**.
64. Liu, S.; Wang, C.; Yin, H.; Molchanov, P.; Wang, Y.F.; Cheng, K.; Chen, M. DoRA: Weight-Decomposed Low-Rank Adaptation. *arXiv preprint arXiv.2402.09353* **2024**.
65. Jie, S.; Deng, Z.H. Convolutional bypasses are better vision transformer adapters. *arXiv preprint arXiv:2207.07039* **2022**.
66. Bai, J.; Chen, D.; Qian, B.; Yao, L.; Li, Y. Federated Fine-tuning of Large Language Models under Heterogeneous Language Tasks and Client Resources. *arXiv preprint arXiv:2402.11505* **2024**.
67. Renduchintala, A.; Konuk, T.; Kuchaiev, O. Tied-LoRA: Enhancing parameter efficiency of LoRA with Weight Tying. In Proceedings of the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2024.
68. Wu, P.; Li, K.; Wang, T.; Wang, F. FedMS: Federated Learning with Mixture of Sparsely Activated Foundations Models. *arXiv preprint arXiv:2312.15926* **2023**.
69. Li, X.L.; Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* **2021**, *abs/2101.00190*.
70. Biderman, D.; Ortiz, J.J.G.; Portes, J.; Paul, M.; Greengard, P.; Jennings, C.; King, D.; Havens, S.; Chiley, V.; Frankle, J.; et al. LoRA Learns Less and Forgets Less. *arXiv preprint arXiv.2405.09673* **2024**.
71. Lee, B.; Park, B.; Kim, C.W.; Ro, Y.M. CoLLaVO: Crayon Large Language and Vision mOdel. *arXiv preprint arXiv:2402.11248* **2024**.
72. Fu, M.; Zhu, K.; Wu, J. Dtl: Disentangled transfer learning for visual recognition. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024.
73. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)* **2017**, *30*.

 doi:10.20944/preprints202504.0743.v1

74. Mujadia, V.; Urlana, A.; Bhaskar, Y.; Pavani, P.A.; Shravya, K.; Krishnamurthy, P.; Sharma, D.M. Assessing Translation Capabilities of Large Language Models Involving English and Indian Languages. *arXiv preprint arXiv:2311.09216* **2023**.
75. Reuther, A.; Michaleas, P.; Jones, M.; Gadepally, V.; Samsi, S.; Kepner, J. Survey and Benchmarking of Machine Learning Accelerators. *2019 IEEE High Performance Extreme Computing Conference (HPEC)* **2019**, pp. 1–9.
76. Gema, A.P.; Daines, L.; Minervini, P.; Alex, B. Parameter-Efficient Fine-Tuning of LLaMA for the Clinical Domain. *arXiv preprint arXiv:2307.03042* **2023**.
77. Zhong, M.; Shen, Y.; Wang, S.; Lu, Y.; Jiao, Y.; Ouyang, S.; Yu, D.; Han, J.; Chen, W. Multi-LoRA Composition for Image Generation. *arXiv preprint arXiv.2402.16843* **2024**.
78. Pan, J.; Lin, Z.; Zhu, X.; Shao, J.; Li, H. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems* **2022**.
79. Chen, A.; Yao, Y.; Chen, P.Y.; Zhang, Y.; Liu, S. Understanding and improving visual prompting: A label-mapping perspective. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19133–19143.
80. Phang, J.; Févry, T.; Bowman, S.R. Sentence Encoders on STILTs: Supplementary Training on Intermediate Labeled-data Tasks. *ArXiv* **2018**, *abs/1811.01088*.
81. Yang, A.X.; Robeyns, M.; Coste, T.; Wang, J.; Bou-Ammar, H.; Aitchison, L. Bayesian Reward Models for LLM Alignment. *arXiv preprint arXiv:2402.13210* **2024**.
82. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* **2018**.
83. Qin, J.; Wu, J.; Yan, P.; Li, M.; Yuxi, R.; Xiao, X.; Wang, Y.; Wang, R.; Wen, S.; Pan, X.; et al. Freeseg: Unified, universal and open-vocabulary image segmentation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19446–19455.
84. Li, J.; Li, D.; Savarese, S.; Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In Proceedings of the International conference on machine learning, 2023.
85. Zhao, Z.; Gan, L.; Wang, G.; Zhou, W.; Yang, H.; Kuang, K.; Wu, F. LoraRetriever: Input-Aware LoRA Retrieval and Composition for Mixed Tasks in the Wild. *arXiv preprint arXiv.2402.09997* **2024**.
86. Zniyed, Y.; Nguyen, T.P.; et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
87. Liu, Y. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* **2019**.
88. Liu, Q.; Wu, X.; Zhao, X.; Zhu, Y.; Xu, D.; Tian, F.; Zheng, Y. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications. *arXiv preprint arXiv:2310.18339* **2023**.
89. Gurrola-Ramos, J.; Dalmau, O.; Alarcón, T.E. A residual dense u-net neural network for image denoising. *IEEE Access* **2021**, *9*, 31742–31754.
90. Lin, Z.; Madotto, A.; Fung, P. Exploring Versatile Generative Language Model Via Parameter-Efficient Transfer Learning. In Proceedings of the Findings, 2020.
91. Wen, Z.; Zhang, J.; Fang, Y. SIBO: A Simple Booster for Parameter-Efficient Fine-Tuning. *arXiv preprint arXiv:2402.11896* **2024**.
92. Guo, D.; Rush, A.M.; Kim, Y. Parameter-Efficient Transfer Learning with Diff Pruning. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2020.
93. Jeon, H.; Kim, Y.; Kim, J.j. L4q: Parameter efficient quantization-aware training on large language models via lora-wise lsq. *arXiv preprint arXiv:2402.04902* **2024**.
94. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. PaLM: Scaling Language Modeling with Pathways. *J. Mach. Learn. Res.* **2023**, *24*, 240:1–240:113.
95. OpenAI. GPT-4 Technical Report. *ArXiv* **2023**, *abs/2303.08774*.
96. Santacroce, M.; Lu, Y.; Yu, H.; Li, Y.; Shen, Y. Efficient RLHF: Reducing the Memory Usage of PPO. *arXiv preprint arXiv:2309.00754* **2023**.
97. Shen, Y.; Xu, Z.; Wang, Q.; Cheng, Y.; Yin, W.; Huang, L. Multimodal Instruction Tuning with Conditional Mixture of LoRA. *arXiv preprint arXiv.2402.15896* **2024**.
98. Sun, S.; Gupta, D.; Iyyer, M. Exploring the impact of low-rank adaptation on the performance, efficiency, and regularization of RLHF. *arXiv preprint arXiv:2309.09055* **2023**.

 doi:10.20944/preprints202504.0743.v1

99. Wu, J.; Li, X.; Wei, C.; Wang, H.; Yuille, A.; Zhou, Y.; Xie, C. Unleashing the power of visual prompting at the pixel level. *arXiv preprint arXiv:2212.10556* **2022**.
100. Zheng, Y.; Zhang, R.; Zhang, J.; Ye, Y.; Luo, Z. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372* **2024**.
101. Kalajdzievski, D. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732* **2023**.
102. Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In Proceedings of the Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 1126–1135.
103. Wang, B.; Wang, W. TDS-CLIP: Temporal Difference Side Network for Image-to-Video Transfer Learning. *arXiv preprint arXiv:2408.10688* **2024**.
104. Sander, M.E.; Ablin, P.; Blondel, M.; Peyré, G. Sinkformers: Transformers with Doubly Stochastic Attention. In Proceedings of the International Conference on Artificial Intelligence and Statistics, 2022, pp. 3515–3530.
105. Fan, T.; Kang, Y.; Ma, G.; Chen, W.; Wei, W.; Fan, L.; Yang, Q. Fate-llm: A industrial grade federated learning framework for large language models. *arXiv preprint arXiv:2310.10049* **2023**.
106. Schneider, S.; Baevski, A.; Collobert, R.; Auli, M. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862* **2019**.
107. Woo, S.; Park, B.; Kim, B.; Jo, M.; Kwon, S.; Jeon, D.; Lee, D. DropBP: Accelerating Fine-Tuning of Large Language Models by Dropping Backward Propagation. *arXiv preprint arXiv:2402.17812* **2024**.
108. Yang, X.; Huang, J.Y.; Zhou, W.; Chen, M. Parameter-Efficient Tuning with Special Token Adaptation. *ArXiv* **2022**, *abs/2210.04382*.
109. Almazrouei, E.; Alobeidli, H.; Alshamsi, A.; Cappelli, A.; Cojocaru, R.; Debbah, M.; Goffinet, É.; Hesslow, D.; Launay, J.; Malartic, Q.; et al. The falcon series of open language models. *arXiv preprint arXiv:2311.16867* **2023**.
110. Huang, C.; Liu, Q.; Lin, B.Y.; Pang, T.; Du, C.; Lin, M. LoraHub: Efficient Cross-Task Generalization via Dynamic LoRA Composition. *arXiv preprint arXiv.2307.13269* **2023**.
111. Tu, M.; Berisha, V.; Woolf, M.; sun Seo, J.; Cao, Y. Ranking the parameters of deep neural networks using the fisher information. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* **2016**, pp. 2647–2651.
112. Feng, W.; Hao, C.; Zhang, Y.; Han, Y.; Wang, H. Mixture-of-LoRAs: An Efficient Multitask Tuning Method for Large Language Models. In Proceedings of the Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, 2024, pp. 11371–11380.
113. Ren, Y.; Zhou, Y.; Yang, J.; Shi, J.; Liu, D.; Liu, F.; Kwon, M.; Shrivastava, A. Customize-a-video: One-shot motion customization of text-to-video diffusion models. *ECCV* **2024**.
114. Liu, M.; Li, B.; Yu, Y. OmniCLIP: Adapting CLIP for Video Recognition with Spatial-Temporal Omni-Scale Feature Learning. *arXiv preprint arXiv:2408.06158* **2024**.
115. Wang, H.; Chang, J.; Zhai, Y.; Luo, X.; Sun, J.; Lin, Z.; Tian, Q. Lion: Implicit vision prompt tuning. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024.
116. Hong, W.; Wang, W.; Ding, M.; Yu, W.; Lv, Q.; Wang, Y.; Cheng, Y.; Huang, S.; Ji, J.; Xue, Z.; et al. CogVLM2: Visual Language Models for Image and Video Understanding. *arXiv preprint arXiv:2408.16500* **2024**.
117. Basu, S.; Hu, S.; Massiceti, D.; Feizi, S. Strong Baselines for Parameter-Efficient Few-Shot Fine-Tuning. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024.
118. Han, Z.; Gao, C.; Liu, J.; Zhang, S.Q.; et al. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608* **2024**.
119. Belofsky, J. Token-Level Adaptation of LoRA Adapters for Downstream Task Generalization. In Proceedings of the 6th Artificial Intelligence and Cloud Computing Conference, 2023, pp. 168–172.
120. Zhang, L.; Zhang, L.; Shi, S.; Chu, X.; Li, B. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303* **2023**.
121. Sidahmed, H.; Phatale, S.; Hutcheson, A.; Lin, Z.; Chen, Z.; Yu, Z.; Jin, J.; Komarytsia, R.; Ahlheim, C.; Zhu, Y.; et al. PERL:Parameter Efficient Reinforcement Learning from Human Feedback. *arXiv preprint arXiv:2403.10704* **2024**.
122. Liao, Q.; Xia, G.; Wang, Z. Calliffusion: Chinese Calligraphy Generation and Style Transfer with Diffusion Modeling. *arXiv preprint arXiv:2305.19124* **2023**.

 doi:10.20944/preprints202504.0743.v1

123. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Proceedings of the Proceedings of the 33nd International Conference on Machine Learning, 2016, pp. 1050–1059.
124. Cheng, J.; Xie, P.; Xia, X.; Li, J.; Wu, J.; Ren, Y.; Li, H.; Xiao, X.; Zheng, M.; Fu, L. ResAdapter: Domain Consistent Resolution Adapter for Diffusion Models. *arXiv preprint arXiv:2403.02084* **2024**.
125. Zhao, J.; Wang, T.; Abid, W.; Angus, G.; Garg, A.; Kinnison, J.; Sherstinsky, A.; Molino, P.; Addair, T.; Rishi, D. LoRA Land: 310 Fine-tuned LLMs that Rival GPT-4, A Technical Report. *arXiv preprint arXiv.2405.00732* **2024**.
126. Sung, Y.L.; Cho, J.; Bansal, M. VL-ADAPTER: Parameter-Efficient Transfer Learning for Vision-and-Language Tasks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* **2021**, pp. 5217–5227.
127. Hu, Y.; Xie, Y.; Wang, T.; Chen, M.; Pan, Z. Structure-Aware Low-Rank Adaptation for Parameter-Efficient Fine-Tuning. *Mathematics* **2023**, *11*, 4317.
128. Li, Y.; Yu, Y.; Liang, C.; He, P.; Karampatziakis, N.; Chen, W.; Zhao, T. Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659* **2023**.
129. Zhang, L.; Zhang, L.; Shi, S.; Chu, X.; Li, B. LoRA-FA: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303* **2023**.
130. Xu, J.; Liu, X.; Wu, Y.; Tong, Y.; Li, Q.; Ding, M.; Tang, J.; Dong, Y. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems* **2024**, *36*.
131. Luo, S.; Tan, Y.; Patil, S.; Gu, D.; von Platen, P.; Passos, A.; Huang, L.; Li, J.; Zhao, H. LCM-LoRA: A Universal Stable-Diffusion Acceleration Module. *arXiv preprint arXiv:2311.05556* **2023**.
132. Ahmad, S.; Chanda, S.; Rawat, Y.S. EZ-CLIP: Efficient Zeroshot Video Action Recognition. *arXiv preprint arXiv:2312.08010* **2023**.
133. Ren, W.; Li, X.; Wang, L.; Zhao, T.; Qin, W. Analyzing and Reducing Catastrophic Forgetting in Parameter Efficient Tuning. *arXiv preprint arXiv.2402.18865* **2024**.
134. Zniyed, Y.; Nguyen, T.P.; et al. Efficient tensor decomposition-based filter pruning. *Neural Networks* **2024**, *178*, 106393.
135. Zi, B.; Qi, X.; Wang, L.; Wang, J.; Wong, K.F.; Zhang, L. Delta-LoRA: Fine-Tuning High-Rank Parameters with the Delta of Low-Rank Matrices. *ArXiv* **2023**, *abs/2309.02411*.
136. Workshop, B.; Scao, T.L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A.S.; Yvon, F.; et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100* **2022**.