Review



# Advances in Parameter-Efficient Fine-Tuning: Optimizing Foundation Models for Scalable AI

Shufen Zhihao *

Posted Date: 27 March 2025

doi: 10.20944/preprints202503.2048.v1

Keywords: parameter-efficient fine-tuning; foundation models; large language models; low-rank adaptation; adapters; prompt tuning; BitFit; federated learning; multimodal learning; edge AI; scalable AI; transfer learning

 doi:10.20944/preprints202503.2048.v1



*Review*

# Advances in Parameter-Efficient Fine-Tuning: Optimizing Foundation Models for Scalable AI

**Shufen Zhihao**

Department of Computer Science and Technology, Fudan University, China; shufen.zhihao@fudan.edu.cn

**Abstract:** The unprecedented scale and capabilities of foundation models, such as large language models and vision transformers, have transformed artificial intelligence (AI) across diverse domains. However, fine-tuning these models for specific tasks remains computationally expensive and memory-intensive, posing challenges for practical deployment, especially in resource-constrained environments. Parameter-efficient fine-tuning (PEFT) methods have emerged as a promising solution, enabling efficient adaptation of large-scale models with minimal parameter updates while maintaining high performance. This survey provides a comprehensive review of PEFT techniques, categorizing existing approaches into adapter-based tuning, low-rank adaptation (LoRA), prefix and prompt tuning, BitFit, and hybrid strategies. We analyze their theoretical foundations, trade-offs in computational efficiency and expressiveness, and empirical performance across various tasks. Furthermore, we explore real-world applications of PEFT in natural language processing, computer vision, multimodal learning, and edge computing, highlighting its impact on accessibility and scalability. Beyond existing methodologies, we discuss emerging trends in PEFT, including meta-learning, dynamic fine-tuning strategies, cross-modal adaptation, and federated fine-tuning. We also address key challenges such as optimal method selection, interpretability, and deployment considerations, paving the way for future research. As foundation models continue to grow, PEFT will remain a crucial area of study, ensuring that the benefits of large-scale AI systems are broadly accessible, efficient, and sustainable.

**Keywords:** parameter-efficient fine-tuning; foundation models; large language models; low-rank adaptation; adapters; prompt tuning; BitFit; federated learning; multimodal learning; edge AI; scalable AI; transfer learning

## 1. Introduction

Foundation models, large-scale neural networks pre-trained on vast amounts of data, have revolutionized the field of artificial intelligence by achieving state-of-the-art performance across a wide range of tasks [1]. These models, such as GPT, BERT, T5, and CLIP, exhibit remarkable generalization capabilities, enabling their application to numerous downstream tasks with minimal adaptation [2]. However, fine-tuning these models for specific applications poses significant challenges, particularly due to their enormous size, high computational costs, and memory-intensive requirements [3]. As a result, researchers have increasingly explored parameter-efficient fine-tuning (PEFT) techniques, which aim to adapt foundation models to new tasks with significantly reduced computational and storage overhead. Traditional fine-tuning approaches involve updating all model parameters to optimize performance on a specific task [4]. While effective, this method is prohibitively expensive for modern foundation models, which often contain billions of parameters. Additionally, full fine-tuning leads to storage inefficiencies, as a separate copy of the model must be maintained for each downstream task. In contrast, PEFT techniques modify only a small subset of parameters while keeping the majority of the pre-trained model fixed [5]. This approach not only reduces computational and memory requirements but also facilitates efficient multi-task learning and continual learning by minimizing catastrophic forgetting [6]. Several PEFT methods have emerged, each leveraging different strategies to achieve efficient adaptation. Low-rank adaptation (LoRA) injects trainable low-rank matrices into specific

 doi:10.20944/preprints202503.2048.v1



layers of the model, enabling efficient weight updates with minimal parameter overhead [7]. Adapter-based methods introduce lightweight, task-specific layers that are fine-tuned while keeping the original model frozen. Prompt tuning and prefix tuning leverage soft prompt embeddings or learnable prefix vectors to steer model behavior without modifying its internal weights [8]. BitFit, a simpler yet effective approach, fine-tunes only the bias terms of the model, significantly reducing the number of trainable parameters [9]. The adoption of PEFT methods has been particularly beneficial in resource-constrained environments, where full fine-tuning is infeasible due to hardware limitations [10]. By reducing the computational cost of adaptation, PEFT enables democratized access to foundation models, empowering researchers and practitioners with limited resources to leverage state-of-the-art AI systems [11]. Moreover, PEFT methods facilitate model deployment in edge devices, where memory and processing power are restricted, thereby broadening the applicability of foundation models beyond traditional cloud-based infrastructure [12]. Despite the advantages of PEFT, several open challenges remain. The trade-offs between efficiency and performance must be carefully considered, as reducing the number of trainable parameters can sometimes lead to suboptimal task adaptation [13]. Moreover, the interpretability and robustness of PEFT methods require further investigation to ensure reliable and consistent outcomes across diverse applications. Additionally, as foundation models continue to scale, new strategies for efficient adaptation must be developed to keep pace with their increasing complexity. This survey provides a comprehensive overview of parameter-efficient fine-tuning techniques for foundation models, examining their methodologies, advantages, limitations, and practical applications [14]. We categorize existing approaches, analyze their effectiveness across various benchmarks, and highlight emerging trends in the field. By consolidating the current state of PEFT research, this survey aims to guide researchers and practitioners in selecting appropriate fine-tuning strategies for their specific needs and to inspire future advancements in efficient model adaptation.

## 2. Background

The emergence of foundation models has significantly transformed the landscape of artificial intelligence, enabling the development of powerful, general-purpose systems that can be adapted to diverse downstream tasks [15]. These models, typically based on deep neural networks, are pre-trained on large-scale corpora using self-supervised or unsupervised learning techniques. Once trained, they serve as a foundation for various applications, requiring only minor modifications or additional fine-tuning to achieve optimal performance on specific tasks [16].

### *2.1. Foundation Models and Transfer Learning*

Foundation models leverage large-scale pre-training to learn generalizable representations of language, vision, and multimodal data [17]. Prominent examples include transformer-based architectures such as BERT [18], GPT [**?** ], T5 [19], and CLIP [17]. These models have demonstrated exceptional performance across a wide range of natural language processing (NLP), computer vision (CV), and multimodal tasks [20]. Their success is largely attributed to their ability to learn hierarchical feature representations from massive datasets, enabling efficient knowledge transfer to new tasks with minimal supervision. Transfer learning plays a crucial role in adapting foundation models to specific applications [21]. In traditional transfer learning, a pre-trained model is fine-tuned on a target dataset by updating its parameters through supervised learning [22]. This approach allows models to retain general knowledge while specializing in domain-specific tasks [23]. However, fine-tuning large-scale models presents significant computational and memory challenges, necessitating the development of more efficient adaptation techniques [24].

### *2.2. Challenges of Full Fine-Tuning*

While full fine-tuning has been the standard approach for adapting foundation models to downstream tasks, it is often impractical for several reasons:

 doi:10.20944/preprints202503.2048.v1



- **Computational Complexity:** Large foundation models contain billions of parameters, making full fine-tuning computationally expensive and requiring high-end hardware resources such as TPUs or GPUs [25].
- **Memory Constraints:** Storing multiple fine-tuned versions of a foundation model for different tasks leads to excessive memory consumption, limiting scalability in real-world applications.
- **Catastrophic Forgetting:** When fine-tuning on new tasks, models often overwrite previously learned knowledge, hindering their ability to retain general capabilities [26].
- **Deployment Challenges:** Fine-tuned models must be stored separately for each task, increasing the cost of deployment and maintenance, particularly in edge computing scenarios [27].

To address these challenges, researchers have developed parameter-efficient fine-tuning (PEFT) methods that reduce the number of trainable parameters while preserving model performance [28]. These methods aim to maintain the benefits of transfer learning while minimizing computational and storage costs [29].

### *2.3. Parameter-Efficient Fine-Tuning: A New Paradigm*

Parameter-efficient fine-tuning (PEFT) has emerged as a promising alternative to full fine-tuning, offering substantial efficiency gains without compromising performance [30]. Instead of updating all model parameters, PEFT techniques modify only a small subset, significantly reducing the number of learnable parameters [31]. This approach offers several advantages:

- **Reduced Computational Cost:** By fine-tuning only a fraction of the model's parameters, PEFT methods lower the training cost, enabling adaptation on resource-constrained devices [32].
- **Improved Storage Efficiency:** Since PEFT methods require storing only a small number of task-specific parameters, they facilitate scalable multi-task learning and efficient model deployment.
- **Modular and Reusable Components:** Many PEFT methods allow task-specific modifications to be applied in a modular fashion, enabling fast adaptation to new domains without extensive retraining [33].
- **Better Transferability:** By preserving the majority of the pre-trained weights, PEFT techniques maintain generalization capabilities, mitigating catastrophic forgetting.

In the following sections, we systematically explore various PEFT methods, categorizing them based on their underlying techniques and assessing their effectiveness across different tasks. We provide a comparative analysis of their computational efficiency, parameter savings, and task performance, highlighting key trends and future research directions in parameter-efficient adaptation [34].

## 3. Taxonomy of Parameter-Efficient Fine-Tuning Methods

Parameter-efficient fine-tuning (PEFT) methods can be broadly categorized based on the underlying strategies they employ to adapt foundation models [35]. These techniques aim to strike a balance between efficiency and task-specific performance by modifying only a small subset of model parameters while leveraging the knowledge stored in the pre-trained weights [36]. In this section, we present a taxonomy of PEFT approaches, classifying them into four major categories: (1) Adapter-Based Methods, (2) Low-Rank Adaptation Methods, (3) Prompt-Based Tuning, and (4) Selective Parameter Tuning [37].

### *3.1. Adapter-Based Methods*

Adapter-based methods introduce additional task-specific modules into the pre-trained model while keeping the original parameters mostly frozen [38]. These methods insert small, trainable neural network layers—known as adapters—into specific parts of the architecture, allowing the model to learn task-relevant transformations without modifying its core parameters [39]. The key advantages of adapter-based methods include modularity, reusability, and efficiency in multi-task learning.

 doi:10.20944/preprints202503.2048.v1



#### 3.1.1. Standard Adapters

The concept of adapters was first proposed in NLP tasks to facilitate efficient transfer learning [40]. A standard adapter consists of a lightweight feed-forward network inserted between layers of a transformer model [41]. These adapters are typically composed of a down-projection layer that reduces the feature dimension, a non-linearity (e.g., ReLU), and an up-projection layer that restores the original dimension [42]. This bottleneck structure ensures that the number of trainable parameters remains minimal.

#### 3.1.2. Compacter and HyperNetwork-Based Adapters

Variants of adapters, such as Compacter [**?** ], further reduce parameter count by leveraging parameterized weight sharing [43]. Instead of maintaining independent adapter weights for each task, these methods use a low-rank reparameterization through hypernetworks, which generate adapter weights dynamically [44]. This approach significantly enhances parameter efficiency while maintaining adaptation flexibility.

### *3.2. Low-Rank Adaptation Methods*

Low-rank adaptation (LoRA) [45] is a highly effective PEFT technique that approximates weight updates using low-rank matrices. Instead of modifying the original model weights, LoRA injects trainable low-rank decomposition matrices into selected layers, capturing essential task-specific variations while keeping the main parameters frozen [46].

#### 3.2.1. LoRA and Its Variants

LoRA operates by decomposing weight updates into a pair of low-rank matrices, reducing the number of trainable parameters by orders of magnitude [47,48]. It allows efficient adaptation with minimal computational overhead [49]. Variants of LoRA, such as QLoRA [**?** ], integrate quantization techniques to further enhance efficiency, making them suitable for large-scale models with memory constraints [50].

### *3.3. Prompt-Based Tuning*

Prompt-based tuning methods leverage additional input tokens or soft embeddings to steer the model's behavior without modifying its internal weights. These approaches include prompt tuning, prefix tuning, and P-Tuning.

#### 3.3.1. Soft Prompt Tuning

Prompt tuning [51] optimizes a small set of trainable prompt vectors that are prepended to the input sequence [52]. These soft prompts act as learnable task-specific instructions, conditioning the model without requiring modifications to its core parameters [53].

#### 3.3.2. Prefix Tuning

Prefix tuning [54] extends soft prompt tuning by introducing learnable prefix embeddings that modulate attention mechanisms at every layer [55]. This technique enables more effective control over the model's internal representations while maintaining a minimal number of trainable parameters.

### *3.4. Selective Parameter Tuning*

Instead of adding new modules or embeddings, selective parameter tuning methods fine-tune only a small subset of existing model parameters [56]. Examples include BitFit [**?** ], which fine-tunes only the bias terms, and Layerwise Fine-Tuning, which selectively updates certain layers.

 doi:10.20944/preprints202503.2048.v1



#### 3.4.1. BitFit

BitFit is an extremely lightweight PEFT method that freezes all weight matrices except for the bias terms [57]. Despite its simplicity, BitFit achieves competitive performance on many NLP tasks while requiring orders of magnitude fewer trainable parameters than full fine-tuning [58].

#### 3.4.2. Layerwise Fine-Tuning

Layerwise fine-tuning approaches selectively update only certain layers of the model, often focusing on higher layers that encode more task-specific knowledge [59]. This strategy balances efficiency and effectiveness, ensuring that the model retains most of its pre-trained knowledge while adapting to new tasks [60].

### *3.5. Comparison of PEFT Methods*

Each PEFT method offers distinct trade-offs in terms of parameter efficiency, computational cost, and task-specific performance. Table 1 summarizes the key characteristics of the major PEFT approaches.

**Table 1.** Comparison of different PEFT methods.

| Method | Trainable Params | Inference Cost | Modularity |
|---|---|---|---|
| Full Fine-Tuning | High | High | No |
| Adapter-Based | Low | Moderate | Yes |
| LoRA | Very Low | Low | No |
| Prompt Tuning | Minimal | Low | Yes |
| BitFit | Minimal | Very Low | No |

In the next section, we delve into an in-depth analysis of the performance of these PEFT techniques across various benchmarks and tasks [61].

## 4. Analysis and Evaluation of PEFT Methods

To understand the effectiveness of parameter-efficient fine-tuning (PEFT) methods, it is essential to evaluate their performance across various benchmarks, considering trade-offs between efficiency and accuracy [62]. This section presents a comparative analysis of PEFT approaches based on multiple criteria, including performance on NLP and vision tasks, computational efficiency, parameter reduction, and generalization capabilities [63].

### *4.1. Performance Across NLP and Vision Benchmarks*

PEFT techniques have been extensively evaluated on a variety of natural language processing (NLP) and computer vision (CV) benchmarks [64]. These evaluations measure the ability of each method to achieve competitive results while significantly reducing the number of trainable parameters.

#### 4.1.1. Natural Language Processing Benchmarks

For NLP tasks, PEFT methods have been assessed on datasets such as:

- **GLUE** [65]: A collection of language understanding tasks including sentiment analysis, textual entailment, and paraphrase detection.
- **SuperGLUE** [**?** ]: An extension of GLUE with more challenging reasoning tasks.
- **SQuAD** [**?** ]: A popular reading comprehension benchmark.
- **MT-Bench** [66]: A benchmark evaluating multi-task generalization across various NLP tasks [67].

Across these benchmarks, adapter-based methods and LoRA have shown strong performance, often achieving results comparable to full fine-tuning while requiring significantly fewer parameters [68]. Prompt tuning methods, while efficient, sometimes exhibit reduced effectiveness on tasks requiring deep syntactic understanding [69].

 doi:10.20944/preprints202503.2048.v1



#### 4.1.2. Computer Vision Benchmarks

In computer vision, PEFT techniques have been evaluated on tasks such as:

- **ImageNet** [**?** ]: A large-scale image classification benchmark [70].
- **COCO** [**?** ]: A dataset for object detection and segmentation [71].
- **CIFAR-10/100** [**?** ]: Standard benchmarks for image classification [72].
- **VTAB** [**?** ]: A comprehensive benchmark for evaluating transfer learning in vision tasks [73].

PEFT approaches such as LoRA and adapter-based tuning have demonstrated strong transferability in vision models like ViTs [74] and CLIP [17,75]. Selective parameter tuning methods, such as BitFit, have been less effective in vision tasks due to the importance of spatial representations that require deeper adaptation [76].

### *4.2. Computational Efficiency and Memory Footprint*

One of the primary motivations for PEFT methods is reducing computational and memory overhead. Table 2 compares different PEFT techniques in terms of their training and inference efficiency [77].

**Table 2.** Comparison of computational efficiency and memory footprint of PEFT methods.

| Method | Trainable Parameters | Memory Footprint | Training Speed |
|---|---|---|---|
| Full Fine-Tuning | 100% | High | Slow |
| Adapters | 1-5% | Moderate | Fast |
| LoRA | <1% | Low | Fast |
| Prompt Tuning | <0.1% | Very Low | Very Fast |
| BitFit | <0.1% | Very Low | Fast |

As shown in Table 2, prompt tuning and BitFit have the lowest computational costs, making them highly efficient for real-time adaptation. However, they sometimes underperform on complex reasoning tasks compared to LoRA and adapter-based methods.

### *4.3. Generalization and Transferability*

Another critical factor in evaluating PEFT methods is their ability to generalize across diverse tasks. Key findings include:

- **Adapters and LoRA** exhibit strong cross-task generalization, making them ideal for multi-task and continual learning [78].
- **Prompt tuning** is highly effective in zero-shot and few-shot learning scenarios but may struggle in domain adaptation [79].
- **BitFit** works well for classification tasks but is less effective for structured prediction tasks such as dependency parsing or named entity recognition.

### *4.4. Summary of Key Findings*

Based on our analysis, we summarize the strengths and limitations of different PEFT methods:

- **Adapters:** Strong task performance and modularity but require additional inference computation.
- **LoRA:** Highly efficient and effective for large models, but may require careful layer selection [80].
- **Prompt Tuning:** Extremely parameter-efficient but less robust for complex tasks [81].
- **BitFit:** Simple and effective for classification but limited in complex reasoning tasks.

In the next section, we explore real-world applications of PEFT methods and discuss their practical deployment in industry and research [82].

## 5. Real-World Applications and Deployment

Parameter-efficient fine-tuning (PEFT) methods have gained significant traction in real-world applications, enabling the adaptation of large foundation models in resource-constrained environments

 doi:10.20944/preprints202503.2048.v1



while maintaining high performance. This section explores various domains where PEFT methods have been successfully deployed, highlighting their impact on natural language processing, computer vision, multimodal learning, and edge computing. We also discuss key deployment considerations, including latency, scalability, and energy efficiency [83].

### *5.1. Natural Language Processing Applications*

PEFT methods have revolutionized NLP applications by allowing the fine-tuning of large language models (LLMs) such as GPT-3, T5, and BERT with minimal computational overhead [84]. Some notable applications include:

#### 5.1.1. Conversational AI and Chatbots

Large-scale chatbots and virtual assistants, such as ChatGPT and Google Assistant, rely on PEFT methods to efficiently adapt to new domains, languages, and user preferences [85]. Adapter-based and LoRA-based fine-tuning techniques enable these models to specialize in customer support, healthcare interactions, and financial services while keeping storage and compute costs low [86].

#### 5.1.2. Machine Translation

In machine translation, PEFT methods facilitate efficient adaptation of multilingual models, such as mT5 and XLM-R, to low-resource languages. Instead of retraining the entire model for a new language, adapter-based approaches and prompt tuning allow for effective cross-lingual transfer with minimal parameter updates [87].

#### 5.1.3. Text Summarization and Information Extraction

News summarization, legal document processing, and biomedical text mining benefit from PEFT by enabling rapid customization of large-scale transformer models for domain-specific tasks [88]. LoRA and prefix-tuning approaches have shown strong results in summarization benchmarks while keeping computational costs manageable [89].

### *5.2. Computer Vision Applications*

In computer vision, PEFT methods have enabled scalable fine-tuning of vision transformers (ViTs) and CLIP-based models for diverse applications, including:

#### 5.2.1. Medical Image Analysis

PEFT methods such as LoRA and adapter tuning have been successfully applied in medical imaging tasks, including disease diagnosis and radiology report generation. For example, fine-tuning pre-trained ViTs on MRI or CT scan datasets with adapter modules allows for efficient deployment in healthcare settings without excessive retraining costs [90].

#### 5.2.2. Autonomous Driving

PEFT techniques help adapt large vision models for real-time perception in autonomous vehicles [91]. Selective parameter tuning methods, such as BitFit, enable efficient updates to object detection and scene segmentation models, allowing adaptation to new environments with minimal retraining.

#### 5.2.3. Retail and E-commerce

Visual recommendation systems and automated product tagging in e-commerce platforms leverage PEFT to customize pre-trained vision models to brand-specific datasets [92]. For instance, LoRA-based fine-tuning allows companies to personalize search and recommendation algorithms without incurring high computational costs.

 doi:10.20944/preprints202503.2048.v1



### *5.3. Multimodal Learning and Vision-Language Models*

PEFT plays a crucial role in fine-tuning multimodal models such as CLIP, Flamingo, and BLIP, which integrate vision and language understanding [93]. Some key applications include:

#### 5.3.1. Content Moderation and Hate Speech Detection

Social media platforms employ multimodal models for detecting harmful content, combining textual and visual cues. PEFT enables these models to adapt to evolving moderation policies and emerging linguistic patterns efficiently [94].

#### 5.3.2. AI-Generated Art and Image Captioning

Creative AI applications, such as DALL·E and Stable Diffusion, benefit from PEFT by allowing users to customize image generation models with style-specific or domain-specific fine-tuning [95]. Prompt tuning and LoRA-based techniques enable rapid adaptation to new artistic styles without modifying the entire model [96].

### *5.4. Edge Computing and Low-Power Devices*

One of the most significant advantages of PEFT is its applicability in edge computing, where deploying full-scale fine-tuned models is infeasible due to hardware constraints. Some real-world examples include:

#### 5.4.1. Smartphones and IoT Devices

Voice assistants, camera enhancements, and real-time language translation on smartphones leverage PEFT to deliver high-performance AI features without excessive power consumption [97]. For instance, prompt tuning allows language models to personalize voice recognition systems on mobile devices [98].

#### 5.4.2. On-Device Personalization

Recommendation engines and predictive typing systems on smartphones use PEFT methods such as BitFit and prefix tuning to adapt to user behavior while ensuring privacy by keeping models locally fine-tuned [99].

#### 5.4.3. Industrial Automation

In manufacturing, PEFT methods are used to fine-tune anomaly detection models for predictive maintenance [100]. Instead of retraining entire models on new machinery data, LoRA-based adaptation enables efficient model updates with minimal computational overhead.

### *5.5. Deployment Considerations and Challenges*

Despite their advantages, deploying PEFT methods in real-world applications requires careful consideration of several factors:

- **Latency and Inference Speed:** While PEFT reduces training costs, some methods (e.g., adapter-based tuning) introduce additional computational overhead at inference time [101]. Strategies such as pruning and quantization can help mitigate this issue.
- **Scalability Across Tasks:** PEFT enables multi-task adaptation, but managing multiple fine-tuned modules (e.g., task-specific adapters) can lead to complexity in large-scale deployments [102].
- **Energy Efficiency:** Reducing the number of trainable parameters lowers energy consumption, making PEFT ideal for sustainable AI development [103]. However, some methods may still require optimization for real-time applications.
- **Security and Privacy:** Deploying PEFT models on personal devices or enterprise environments raises security concerns [104]. Techniques such as federated learning and encrypted fine-tuning can enhance privacy [43].

 doi:10.20944/preprints202503.2048.v1



### *5.6. Summary and Future Directions*

The real-world adoption of PEFT methods has demonstrated their effectiveness in reducing computational costs while maintaining competitive performance. Key takeaways include:

- PEFT methods are widely used in NLP, CV, and multimodal applications, enabling efficient fine-tuning of foundation models.
- Edge computing benefits significantly from PEFT by enabling AI-driven applications on resource-constrained devices.
- Deployment challenges, such as inference overhead and scalability, require further research to optimize PEFT for production use.

In the next section, we discuss emerging trends and future research directions in PEFT, exploring novel techniques and potential advancements [105].

## 6. Emerging Trends and Future Directions

As foundation models continue to grow in scale and complexity, parameter-efficient fine-tuning (PEFT) remains a crucial area of research [106]. Recent advancements in model architectures, training strategies, and optimization techniques are shaping the future of PEFT [107]. In this section, we explore emerging trends and outline potential future directions, including advanced low-rank adaptation methods, dynamic fine-tuning strategies, cross-modal adaptation, and efficient deployment in decentralized and federated settings [108].

### *6.1. Advanced Low-Rank Adaptation Techniques*

Low-rank adaptation (LoRA) has proven to be one of the most effective PEFT methods, but ongoing research aims to enhance its flexibility and efficiency [109]. Key developments include:

- **Dynamic Low-Rank Updates:** Instead of using a fixed low-rank decomposition, recent approaches propose dynamically adjusting the rank of adaptation matrices based on task complexity, leading to more efficient fine-tuning [110].
- **Sparse Low-Rank Decomposition:** Hybrid approaches that combine sparsity with low-rank updates aim to further reduce the number of trainable parameters while maintaining model expressiveness [111].
- **Rank Selection Optimization:** Methods such as AutoLoRA explore automatic selection of optimal rank values, reducing the need for manual hyperparameter tuning.

### *6.2. Meta-Learning and Dynamic Fine-Tuning*

One of the limitations of current PEFT techniques is that they require separate tuning for each new task [112]. Emerging approaches in meta-learning and dynamic adaptation aim to mitigate this:

- **Task-Agnostic Fine-Tuning:** Instead of learning separate parameters for each task, meta-learning-based PEFT approaches enable models to generalize to unseen tasks with minimal updates [113].
- **Adaptive Parameter-Freezing:** Future PEFT techniques may incorporate automated freezing and unfreezing of parameters based on task complexity, optimizing the trade-off between efficiency and performance [114].
- **Few-Shot and Continual Learning:** Integrating PEFT with few-shot and continual learning paradigms allows models to efficiently acquire new knowledge while avoiding catastrophic forgetting.

### *6.3. Cross-Modal and Multilingual Adaptation*

Foundation models are increasingly trained on diverse modalities, necessitating efficient adaptation techniques that extend beyond a single modality or language:

 doi:10.20944/preprints202503.2048.v1



- **Unified Multimodal PEFT:** Research is exploring ways to integrate PEFT across vision-language, speech-text, and other multimodal domains, allowing a single fine-tuned module to operate across different inputs [115].
- **Cross-Lingual Adaptation:** PEFT methods that enable seamless adaptation of multilingual models across new languages, especially for low-resource settings, are gaining attention [116].
- **Domain-Agnostic PEFT:** Instead of learning separate PEFT modules for each domain (e.g., medical vs [117]. legal text), future approaches may develop universal PEFT methods that generalize across domains [118].

### *6.4. Federated and Decentralized Fine-Tuning*

With growing concerns over data privacy and computational efficiency, federated and decentralized fine-tuning approaches are becoming critical areas of research:

- **Federated PEFT:** Techniques such as federated LoRA allow models to be fine-tuned across decentralized devices while preserving user privacy [119].
- **On-Device Adaptation:** Lightweight PEFT models optimized for deployment on smartphones, IoT devices, and edge servers reduce reliance on centralized cloud computing [120].
- **Privacy-Preserving Fine-Tuning:** Encryption-based techniques such as homomorphic encryption and secure multi-party computation are being integrated with PEFT to ensure data security during adaptation [121].

### *6.5. Scalable and Hardware-Aware PEFT*

Efficient fine-tuning must also consider hardware constraints, particularly for deployment on accelerators such as GPUs, TPUs, and specialized AI chips:

- **Quantized and Pruned PEFT Models:** Applying quantization and pruning to PEFT methods can further reduce memory and inference costs without compromising accuracy [122].
- **PEFT for Sparse Models:** Sparse models such as mixture-of-experts (MoE) architectures require specialized PEFT techniques that adapt only active model components [123].
- **Accelerator-Aware Fine-Tuning:** Future PEFT approaches will need to optimize for hardware-specific characteristics, such as tensor core efficiency on GPUs or reduced precision arithmetic on edge AI chips.

### *6.6. Bridging PEFT with Neural Architecture Search*

Neural Architecture Search (NAS) has revolutionized the discovery of efficient deep learning architectures [124]. A promising future direction is integrating NAS with PEFT to automatically identify the most efficient fine-tuning configurations:

- **Task-Specific PEFT Search:** Automating the selection of PEFT methods based on task characteristics to optimize efficiency and accuracy [125].
- **Hybrid NAS-PEFT Approaches:** Exploring how NAS can guide the design of efficient adapters, LoRA layers, or prompt tuning strategies for foundation models [126].
- **Energy-Efficient NAS-PEFT Pipelines:** Combining NAS with PEFT to identify configurations that minimize energy consumption while maintaining high performance [127].

### *6.7. Challenges and Open Questions*

Despite the rapid progress in PEFT research, several open challenges remain:

- **Optimal PEFT Selection:** How can models automatically determine which PEFT method is best suited for a given task?
- **Interpretability:** How do PEFT modifications affect the internal representations of foundation models, and can they be made more interpretable [128]?
- **Scalability to Extremely Large Models:** As models surpass trillions of parameters, will current PEFT techniques remain viable, or will entirely new methods be needed [129]?

 doi:10.20944/preprints202503.2048.v1



- **Generalization Across Domains:** Can a single PEFT module generalize across multiple domains without requiring separate fine-tuning?

### *6.8. Summary and Outlook*

PEFT has become an essential component of modern AI, enabling efficient adaptation of foundation models for a wide range of applications. Looking ahead:

- Research is moving toward more dynamic and adaptive PEFT methods that optimize parameter selection and task generalization [130].
- Multimodal, multilingual, and federated fine-tuning approaches will expand the reach of PEFT beyond traditional NLP and vision tasks.
- Hardware-aware and NAS-integrated PEFT methods will drive new breakthroughs in efficiency and deployment scalability.

As AI systems continue to evolve, PEFT will remain a key enabler of practical and efficient foundation model adaptation, ensuring that powerful AI capabilities remain accessible across diverse domains and computational environments.

## 7. Conclusion

The rapid evolution of foundation models has necessitated the development of parameter-efficient fine-tuning (PEFT) techniques, allowing for scalable adaptation without incurring prohibitive computational and memory costs. This survey has explored the fundamental principles, methodologies, real-world applications, and emerging trends in PEFT, highlighting its critical role in democratizing access to large-scale AI models.

### *7.1. Key Takeaways*

Throughout this survey, we have identified several key insights:

- PEFT methods such as adapters, LoRA, prefix tuning, and BitFit offer efficient alternatives to full fine-tuning, significantly reducing the number of trainable parameters while maintaining competitive performance.
- The application of PEFT extends beyond natural language processing to computer vision, multimodal learning, and edge computing, demonstrating its versatility across different AI domains.
- Real-world deployment of PEFT requires careful consideration of inference latency, scalability, energy efficiency, and privacy, motivating ongoing research in optimized architectures.
- Emerging trends such as dynamic fine-tuning, federated adaptation, and hardware-aware optimization are paving the way for more flexible and scalable PEFT strategies.
- Despite its advantages, PEFT still faces challenges related to optimal method selection, interpretability, and generalization across domains, indicating promising directions for future research.

### *7.2. The Future of PEFT*

Looking ahead, PEFT is expected to play an increasingly important role in the AI ecosystem. As models grow even larger, the need for efficient fine-tuning strategies will become even more pronounced, especially in decentralized and privacy-sensitive applications. Future research will likely focus on:

- **Hybrid PEFT Approaches:** Combining different fine-tuning techniques (e.g., LoRA with prompt tuning) to optimize both efficiency and generalization.
- **Task-Agnostic Adaptation:** Developing PEFT methods that allow for seamless cross-domain and cross-modal transfer without requiring extensive task-specific fine-tuning.
- **Neural Architecture Search (NAS) for PEFT:** Automating the discovery of optimal fine-tuning configurations tailored to specific tasks and computational constraints.
- **Sustainable AI:** Further reducing the carbon footprint of model adaptation through lightweight PEFT methods that minimize energy consumption.

 doi:10.20944/preprints202503.2048.v1



### 7.3. Final Thoughts

As AI continues to integrate into various aspects of society, ensuring the accessibility and efficiency of foundation models remains a crucial challenge. PEFT methods provide a promising pathway to making these models more adaptable, cost-effective, and widely deployable. By addressing the open questions and exploring novel approaches, future research in PEFT will contribute to the broader goal of making AI both powerful and sustainable.

## References


1. Xia, X.; Zhang, D.; Liao, Z.; Hou, Z.; Sun, T.; Li, J.; Fu, L.; Dong, Y. SceneGenAgent: Precise Industrial Scene Generation with Coding Agent. *arXiv preprint arXiv:2410.21909* **2024**.
2. Liu, Z.; Li, S.; Luo, Y.; Fei, H.; Cao, Y.; Kawaguchi, K.; Wang, X.; Chua, T. MolCA: Molecular Graph-Language Modeling with Cross-Modal Projector and Uni-Modal Adapter. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 15623–15638.
3. Marjit, S.; Singh, H.; Mathur, N.; Paul, S.; Yu, C.M.; Chen, P.Y. DiffuseKronA: A Parameter Efficient Fine-tuning Method for Personalized Diffusion Model. *arXiv preprint arXiv:2402.17412* **2024**.
4. Davison, J. Compacter: Efficient Low-Rank Hypercomplex Adapter Layers. In Proceedings of the Neural Information Processing Systems, 2021.
5. Silva, A.; Fang, S.; Monperrus, M. Repairllama: Efficient representations and fine-tuned adapters for program repair. *arXiv preprint arXiv:2312.15698* **2023**.
6. Zhang, D.; Yu, Y.; Li, C.; Dong, J.; Su, D.; Chu, C.; Yu, D. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601* **2024**.
7. Yuan, H.; Yuan, Z.; Gan, R.; Zhang, J.; Xie, Y.; Yu, S. BioBART: Pretraining and Evaluation of A Biomedical Generative Language Model. In Proceedings of the Workshop on Biomedical Natural Language Processing, 2022.
8. Lee, A.N.; Hunter, C.J.; Ruiz, N. Platypus: Quick, Cheap, and Powerful Refinement of LLMs. *arXiv preprint arXiv:2308.07317* **2023**.
9. Wu, Y.; Xiang, Y.; Huo, S.; Gong, Y.; Liang, P. LoRA-SP: streamlined partial parameter adaptation for resource efficient fine-tuning of large language models. In Proceedings of the Third International Conference on Algorithms, Microchips, and Network Applications, 2024, pp. 488–496.
10. Dong, W.; Xue, S.; Duan, X.; Han, S. Prompt tuning inversion for text-driven image editing using diffusion models. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023.
11. Zhang, L.; Wu, J.; Zhou, D.; Xu, G. STAR: Constraint LoRA with Dynamic Active Learning for Data-Efficient Fine-Tuning of Large Language Models. *arXiv preprint arXiv.2403.01165* **2024**.
12. 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.
13. Liu, T.; Low, B.K.H. Goat: Fine-tuned LLaMA Outperforms GPT-4 on Arithmetic Tasks. *arXiv preprint arXiv:2305.14201* **2023**.
14. 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**.
15. Yang, A.X.; Robeyns, M.; Wang, X.; Aitchison, L. Bayesian low-rank adaptation for large language models. *arXiv preprint arXiv.2308.13111* **2023**.
16. Pfeiffer, J.; Kamath, A.; Rücklé, A.; Cho, K.; Gurevych, I. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. *ArXiv* **2020**, *abs/2005.00247*.
17. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International conference on machine learning. PMLR, 2021, pp. 8748–8763.
18. Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* **2018**.
19. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* **2020**, *21*, 1–67.

 doi:10.20944/preprints202503.2048.v1




20. 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.
21. Konstantinidis, T.; Iacovides, G.; Xu, M.; Constantinides, T.G.; Mandic, D.P. FinLlama: Financial Sentiment Classification for Algorithmic Trading Applications. *arXiv preprint arXiv:2403.12285* **2024**.
22. Radev, D.R.; Zhang, R.; Rau, A.; Sivaprasad, A.; Hsieh, C.H.; Rajani, N.; Tang, X.; Vyas, A.; Verma, N.; Krishna, P.; et al. DART: Open-Domain Structured Data Record to Text Generation. *ArXiv* **2020**, *abs/2007.02871*.
23. 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.
24. Ren, Y.; Chen, Y.; Liu, S.; Wang, B.; Yu, H.; Cui, Z. TPLLM: A Traffic Prediction Framework Based on Pretrained Large Language Models. *arXiv preprint arXiv:2403.02221* **2024**.
25. Guo, Y.; Yang, C.; Rao, A.; Wang, Y.; Qiao, Y.; Lin, D.; Dai, B. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725* **2023**.
26. Radford, A. Improving language understanding by generative pre-training. *OpenAI* **2018**.
27. Zhang, K.; Liu, D. Customized Segment Anything Model for Medical Image Segmentation. *arXiv preprint arXiv:2304.13785* **2023**.
28. Silva-Rodriguez, J.; Hajimiri, S.; Ben Ayed, I.; Dolz, J. A closer look at the few-shot adaptation of large vision-language models. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024.
29. 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**.
30. 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**.
31. 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**.
32. Chen, Y.; Qian, S.; Tang, H.; Lai, X.; Liu, Z.; Han, S.; Jia, J. LongLoRA: Efficient Fine-tuning of Long-Context Large Language Models. *arXiv preprint arXiv.2309.12307* **2023**.
33. Yao, T.; Pan, Y.; Li, Y.; Qiu, Z.; Mei, T. Boosting image captioning with attributes. In Proceedings of the Proceedings of the IEEE international conference on computer vision, 2017, pp. 4894–4902.
34. Qi, Z.; Tan, X.; Shi, S.; Qu, C.; Xu, Y.; Qi, Y. PILLOW: Enhancing Efficient Instruction Fine-tuning via Prompt Matching. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, 2023, pp. 471–482.
35. Yin, D.; Hu, L.; Li, B.; Zhang, Y.; Yang, X. 5%> 100%: Breaking Performance Shackles of Full Fine-Tuning on Visual Recognition Tasks. *arXiv preprint arXiv:2408.08345* **2024**.
36. Jia, M.; Tang, L.; Chen, B.C.; Cardie, C.; Belongie, S.; Hariharan, B.; Lim, S.N. Visual prompt tuning. In Proceedings of the European Conference on Computer Vision, 2022.
37. Sui, Y.; Yin, M.; Gong, Y.; Xiao, J.; Phan, H.; Yuan, B. ELRT: Efficient Low-Rank Training for Compact Convolutional Neural Networks. *arXiv preprint arXiv.2401.10341* **2024**.
38. 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.
39. 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**.
40. Malladi, S.; Wettig, A.; Yu, D.; Chen, D.; Arora, S. A Kernel-Based View of Language Model Fine-Tuning. In Proceedings of the International Conference on Machine Learning, 2023, pp. 23610–23641.
41. 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.
42. Ye, H.; Zhang, J.; Liu, S.; Han, X.; Yang, W. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721* **2023**.
43. 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.

 doi:10.20944/preprints202503.2048.v1




44. Gu, Y.; Wang, X.; Wu, J.Z.; Shi, Y.; Chen, Y.; Fan, Z.; Xiao, W.; Zhao, R.; Chang, S.; Wu, W.; et al. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. *Advances in Neural Information Processing Systems* **2024**.
45. Hu, E.; Tsai, Y.; Hsu, K.; Sussman, A.; Eleni, K.; Martinez, B.; Martinez, F.; Charle, R. LoRA: Low-Rank Adaptation of Large Language Models. In Proceedings of the Proceedings of the 38th International Conference on Machine Learning (ICML), 2021.
46. Chen, T.; Ding, T.; Yadav, B.; Zharkov, I.; Liang, L. Lorashear: Efficient large language model structured pruning and knowledge recovery. *arXiv preprint arXiv:2310.18356* **2023**.
47. Hu, S.; Liao, Z.; Xia, Y. ProSFDA: prompt learning based source-free domain adaptation for medical image segmentation. *arXiv preprint arXiv:2211.11514* **2022**.
48. Zniyed, Y.; Nguyen, T.P.; et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
49. Mao, Y.; Mathias, L.; Hou, R.; Almahairi, A.; Ma, H.; Han, J.; tau Yih, W.; Khabsa, M. UniPELT: A Unified Framework for Parameter-Efficient Language Model Tuning. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2021.
50. Liu, Y. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* **2019**.
51. Lester, B.; Al-Rfou, R.; Constant, N. The power of scale for parameter-efficient prompt tuning. In Proceedings of the Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2021.
52. Li, Y.; Liu, H.; Wu, Q.; Mu, F.; Yang, J.; Gao, J.; Li, C.; Lee, Y.J. Gligen: Open-set grounded text-to-image generation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023.
53. Liu, S.Y.; Wang, C.Y.; Yin, H.; Molchanov, P.; Wang, Y.C.F.; Cheng, K.T.; Chen, M.H. DoRA: Weight-Decomposed Low-Rank Adaptation. *arXiv preprint arXiv:2402.09353* **2024**.
54. Li, X.; Liang, P.; Zhang, Z.; Xie, D.; Li, Y.; Zhang, Y.; Zhang, H.; Wang, Y.; Wang, R. Prefix-tuning: Optimizing continuous prompts for generation. In Proceedings of the Proceedings of the 39th International Conference on Machine Learning (ICML), 2021.
55. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys* **2021**, *55*, 1 – 35.
56. Zhu, J.; Lai, S.; Chen, X.; Wang, D.; Lu, H. Visual prompt multi-modal tracking. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023.
57. Wang, C.; Duan, Z.; Liu, B.; Zou, X.; Chen, C.; Jia, K.; Huang, J. PAI-Diffusion: Constructing and Serving a Family of Open Chinese Diffusion Models for Text-to-image Synthesis on the Cloud. *arXiv preprint arXiv:2309.05534* **2023**.
58. Ansell, A.; Ponti, E.; Pfeiffer, J.; Ruder, S.; Glavas, G.; Vulic, I.; Korhonen, A. MAD-G: Multilingual Adapter Generation for Efficient Cross-Lingual Transfer. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2021.
59. Dong, B.; Zhou, P.; Yan, S.; Zuo, W. LPT: Long-tailed prompt tuning for image classification. *International Journal of Computer Vision* **2023**.
60. Li, S. DiffStyler: Diffusion-based Localized Image Style Transfer. *arXiv preprint arXiv:2403.18461* **2024**.
61. Liu, K.L.; Li, W.J.; Guo, M. Emoticon smoothed language models for twitter sentiment analysis. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2012, Vol. 26, pp. 1678–1684.
62. Yang, S.; Ali, M.A.; Wang, C.; Hu, L.; Wang, D. MoRAL: MoE Augmented LoRA for LLMs' Lifelong Learning. *arXiv preprint arXiv.2402.11260* **2024**.
63. Ayupov, S.; Chirkova, N. Parameter-efficient finetuning of transformers for source code. *arXiv preprint arXiv:2212.05901* **2022**.
64. Zhao, B.; Hajishirzi, H.; Cao, Q. Apt: Adaptive pruning and tuning pretrained language models for efficient training and inference. *arXiv preprint arXiv:2401.12200* **2024**.
65. 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**.
66. Zheng, L.; Chiang, W.L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.P.; et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena, 2023, [arXiv:cs.CL/2306.05685].
67. 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.

 doi:10.20944/preprints202503.2048.v1




68. Liu, Z.; Lyn, J.; Zhu, W.; Tian, X.; Graham, Y. ALoRA: Allocating Low-Rank Adaptation for Fine-tuning Large Language Models. *arXiv preprint arXiv.2403.16187* **2024**.
69. Ye, M.; Fang, X.; Du, B.; Yuen, P.C.; Tao, D. Heterogeneous Federated Learning: State-of-the-art and Research Challenges. *ACM Computing Surveys* **2024**, *56*, 79:1–79:44.
70. 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**.
71. 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**.
72. Yao, H.; Wu, W.; Li, Z. Side4video: Spatial-temporal side network for memory-efficient image-to-video transfer learning. *arXiv preprint arXiv:2311.15769* **2023**.
73. Mangrulkar, S.; Gugger, S.; Debut, L.; Belkada, Y.; Paul, S.; Bossan, B. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft **2022**.
74. DOSOVITSKIY, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* **2020**.
75. Yoo, S.; Kim, K.; Kim, V.G.; Sung, M. As-Plausible-As-Possible: Plausibility-Aware Mesh Deformation Using 2D Diffusion Priors. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 4315–4324.
76. Zniyed, Y.; Nguyen, T.P.; et al. Efficient tensor decomposition-based filter pruning. *Neural Networks* **2024**, *178*, 106393.
77. Golnari, P.A. LoRA-Enhanced Distillation on Guided Diffusion Models. *arXiv preprint arXiv:2312.06899* **2023**.
78. Zhu, Y.; Wichers, N.; Lin, C.; Wang, X.; Chen, T.; Shu, L.; Lu, H.; Liu, C.; Luo, L.; Chen, J.; et al. SiRA: Sparse Mixture of Low Rank Adaptation. *arXiv preprint arXiv.2311.09179* **2023**.
79. Xu, R.; Luo, F.; Zhang, Z.; Tan, C.; Chang, B.; Huang, S.; Huang, F. Raise a Child in Large Language Model: Towards Effective and Generalizable Fine-tuning. *ArXiv* **2021**, *abs/2109.05687*.
80. 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.
81. Suri, K.; Mishra, P.; Saha, S.; Singh, A. SuryaKiran at MEDIQA-Sum 2023: Leveraging LoRA for Clinical Dialogue Summarization. In Proceedings of the Working Notes of the Conference and Labs of the Evaluation Forum, 2023, pp. 1720–1735.
82. Ding, N.; Qin, Y.; Yang, G.; Wei, F.; Yang, Z.; Su, Y.; Hu, S.; Chen, Y.; Chan, C.; Chen, W.; et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nat. Mac. Intell.* **2023**, *5*, 220–235.
83. Huang, T.; Zeng, Y.; Zhang, Z.; Xu, W.; Xu, H.; Xu, S.; Lau, R.W.H.; Zuo, W. DreamControl: Control-Based Text-to-3D Generation with 3D Self-Prior. *arXiv preprint arXiv:2312.06439* **2023**.
84. Zhu, Y.; Yang, X.; Wu, Y.; Zhang, W. Parameter-efficient fine-tuning with layer pruning on free-text sequence-to-sequence modeling. *arXiv preprint arXiv:2305.08285* **2023**.
85. 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**.
86. Zhang, J.O.; Sax, A.; Zamir, A.; Guibas, L.; Malik, J. Side-tuning: a baseline for network adaptation via additive side networks. In Proceedings of the European Conference on Computer Vision, 2020.
87. Wang, H.; Xiao, Z.; Li, Y.; Wang, S.; Chen, G.; Chen, Y. MiLoRA: Harnessing Minor Singular Components for Parameter-Efficient LLM Finetuning. *arXiv preprint arXiv:2406.09044* **2024**.
88. Ge, Y.; Ge, Y.; Zeng, Z.; Wang, X.; Shan, Y. Planting a SEED of Vision in Large Language Model. *arXiv preprint arXiv:2307.08041* **2023**.
89. Tran, H.; Yang, Z.; Yao, Z.; Yu, H. BioInstruct: Instruction Tuning of Large Language Models for Biomedical Natural Language Processing. *arXiv preprint arXiv:2310.19975* **2023**.
90. Frank, M.; Wolfe, P.; et al. An algorithm for quadratic programming. *Naval research logistics quarterly* **1956**, *3*, 95–110.
91. 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**.
92. Liu, G.; Xia, M.; Zhang, Y.; Chen, H.; Xing, J.; Wang, X.; Yang, Y.; Shan, Y. Stylecrafter: Enhancing stylized text-to-video generation with style adapter. *arXiv preprint arXiv:2312.00330* **2023**.
93. Kovaleva, O.; Romanov, A.; Rogers, A.; Rumshisky, A. Revealing the Dark Secrets of BERT. *ArXiv* **2019**, *abs/1908.08593*.

 doi:10.20944/preprints202503.2048.v1




94. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In Proceedings of the European Conference on Computer Vision, 2016.
95. Aghajanyan, A.; Zettlemoyer, L.; Gupta, S. Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2020.
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. Ye, Z.; Lovell, L.; Faramarzi, A.; Ninic, J. SAM-based instance segmentation models for the automation of structural damage detection. *arXiv preprint arXiv:2401.15266* **2024**.
98. Shin, T.; Razeghi, Y.; IV, R.L.L.; Wallace, E.; Singh, S. Eliciting Knowledge from Language Models Using Automatically Generated Prompts. *ArXiv* **2020**, *abs/2010.15980*.
99. Tan, W.; Zhang, W.; Liu, S.; Zheng, L.; Wang, X.; An, B. True knowledge comes from practice: Aligning llms with embodied environments via reinforcement learning. *arXiv preprint arXiv:2401.14151* **2024**.
100. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; chun Woo, W. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In Proceedings of the NIPS, 2015.
101. Tang, Z.; Yang, Z.; Zhu, C.; Zeng, M.; Bansal, M. Any-to-any generation via composable diffusion. *Advances in Neural Information Processing Systems* **2024**.
102. Mu, Y.; Zhang, Q.; Hu, M.; Wang, W.; Ding, M.; Jin, J.; Wang, B.; Dai, J.; Qiao, Y.; Luo, P. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *Advances in Neural Information Processing Systems* **2024**, *36*.
103. Hong, W.; Ding, M.; Zheng, W.; Liu, X.; Tang, J. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868* **2022**.
104. Wen, Y.; Chaudhuri, S. Batched Low-Rank Adaptation of Foundation Models. *arXiv preprint arXiv.2312.05677* **2023**.
105. Wu, J.Z.; Ge, Y.; Wang, X.; Lei, S.W.; Gu, Y.; Shi, Y.; Hsu, W.; Shan, Y.; Qie, X.; Shou, M.Z. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023.
106. 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* **2023**, *abs/2303.18223*.
107. Rücklé, A.; Geigle, G.; Glockner, M.; Beck, T.; Pfeiffer, J.; Reimers, N.; Gurevych, I. AdapterDrop: On the Efficiency of Adapters in Transformers. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2020.
108. Zhang, N.; Bi, Z.; Liang, X.; Cheng, S.; Hong, H.; Deng, S.; Lian, J.; Zhang, Q.; Chen, H. OntoProtein: Protein Pretraining With Gene Ontology Embedding. *ArXiv* **2022**, *abs/2201.11147*.
109. Liu, S.Y.; Wang, C.Y.; Yin, H.; Molchanov, P.; Wang, Y.C.F.; Cheng, K.T.; Chen, M.H. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353* **2024**.
110. Lin, H.; Cho, J.; Zala, A.; Bansal, M. Ctrl-Adapter: An Efficient and Versatile Framework for Adapting Diverse Controls to Any Diffusion Model. *arXiv preprint arXiv:2404.09967* **2024**.
111. 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.
112. Li, S.; Lu, H.; Wu, T.; Yu, M.; Weng, Q.; Chen, X.; Shan, Y.; Yuan, B.; Wang, W. CaraServe: CPU-Assisted and Rank-Aware LoRA Serving for Generative LLM Inference. *arXiv preprint arXiv:2401.11240* **2024**.
113. Conneau, A.; Lample, G. Cross-lingual language model pretraining. *Advances in neural information processing systems* **2019**, *32*.
114. Luo, Z.; Xu, X.; Liu, F.; Koh, Y.S.; Wang, D.; Zhang, J. Privacy-Preserving Low-Rank Adaptation for Latent Diffusion Models. *arXiv preprint arXiv.2402.11989* **2024**.
115. Gardent, C.; Shimorina, A.; Narayan, S.; Perez-Beltrachini, L. The WebNLG Challenge: Generating Text from RDF Data. In Proceedings of the International Conference on Natural Language Generation, 2017.
116. Jiang, Z.; Chen, T.; Chen, X.; Cheng, Y.; Zhou, L.; Yuan, L.; Awadallah, A.; Wang, Z. Dna: Improving few-shot transfer learning with low-rank decomposition and alignment. In Proceedings of the European Conference on Computer Vision, 2022.
117. Chen, Z.; Huang, H.; Andrusenko, A.; Hrinchuk, O.; Puvvada, K.C.; Li, J.; Ghosh, S.; Balam, J.; Ginsburg, B. SALM: Speech-augmented Language Model with In-context Learning for Speech Recognition and Translation. *arXiv preprint arXiv:2310.09424* **2023**.

 doi:10.20944/preprints202503.2048.v1




118. Li, J.; Li, D.; Xiong, C.; Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In Proceedings of the International conference on machine learning. PMLR, 2022.
119. Bu, Z.; Wang, Y.X.; Zha, S.; Karypis, G. Differentially private bias-term only fine-tuning of foundation models. *Advances in Neural Information Processing Systems* **2022**.
120. Koubbi, H.; Boussard, M.; Hernandez, L. The Impact of LoRA on the Emergence of Clusters in Transformers. *arXiv preprint arXiv.2402.15415* **2024**.
121. Sun, Y.; Li, Z.; Li, Y.; Ding, B. Improving LoRA in Privacy-preserving Federated Learning. *arXiv preprint arXiv:2403.12313* **2024**.
122. Houlsby, N.; Giurgiu, A.; Hu, X.; Goyal, N.; Rashid, A.; Vaswani, A.; Shazeer, N. Parameter-efficient transfer learning for NLP. In Proceedings of the Proceedings of the 36th International Conference on Machine Learning (ICML), 2019.
123. Ding, N.; Qin, Y.; Yang, G.; Wei, F.; Yang, Z.; Su, Y.; Hu, S.; Chen, Y.; Chan, C.M.; Chen, W.; et al. Delta Tuning: A Comprehensive Study of Parameter Efficient Methods for Pre-trained Language Models. *ArXiv* **2022**, *abs/2203.06904*.
124. Zhang, Y.; Xu, Q.; Zhang, L. DragTex: Generative Point-Based Texture Editing on 3D Mesh. *arXiv preprint arXiv:2403.02217* **2024**.
125. 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**.
126. Li, Z.; Wang, Y.; Zhi, T.; Chen, T. A survey of neural network accelerators. *Frontiers of Computer Science* **2017**, *11*, 746–761.
127. 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**.
128. Tworkowski, S.; Staniszewski, K.; Pacek, M.; Wu, Y.; Michalewski, H.; Milos, P. Focused Transformer: Contrastive Training for Context Scaling. In Proceedings of the Advances in Neural Information Processing Systems, 2023.
129. Xing, J.; Wang, M.; Hou, X.; Dai, G.; Wang, J.; Liu, Y. Multimodal adaptation of clip for few-shot action recognition. *arXiv preprint arXiv:2308.01532* **2023**.
130. Dettmers, T.; Pagnoni, A.; Holtzman, A.; Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. In Proceedings of the Advances in Neural Information Processing Systems, 2024.