Preprints.org

Review

Not peer-reviewed version

Fine-Tuning Transformers Efficiently: A Survey on LoRA and Its Impact

Muchen Huan and Jianhong Shun *

Posted Date: 20 February 2025

doi: 10.20944/preprints202502.1637.v1

Keywords: low-rank adaptation; large language models; parameter-efficient fine-tuning; deep learning; transfer learning

Review

Fine-Tuning Transformers Efficiently: A Survey on LoRA and Its Impact

Muchen Huan and Jianhong Shun *

The Frontier Institute of Science and Technology (FIST), Xi'an Jiaotong University, China
* Correspondence: jianhong.shun@mail.xjtu.edu.cn

Abstract: The rapid growth of Large Language Models (LLMs) has revolutionized natural language processing (NLP), enabling remarkable advancements in text generation, machine translation, and various downstream applications. However, fine-tuning these models remains computationally expensive due to their vast number of parameters. Low-Rank Adaptation (LoRA) has emerged as a highly efficient parameter-efficient fine-tuning (PEFT) technique that significantly reduces memory and computational costs while maintaining competitive performance. LoRA achieves this by freezing the pre-trained model weights and introducing trainable low-rank matrices into transformer layers, enabling efficient adaptation to new tasks. This survey provides a comprehensive review of LoRA, covering its theoretical foundations, practical implementation, recent advancements, and real-world applications. We explore various hybrid approaches that combine LoRA with other fine-tuning techniques, such as prompt tuning and adapter layers, as well as extensions like dynamic rank selection and quantized LoRA for enhanced efficiency. Additionally, we discuss the application of LoRA beyond traditional NLP tasks, including vision-language models, speech processing, and reinforcement learning. Despite its advantages, LoRA presents challenges such as inference overhead and optimal rank selection, which remain active areas of research. We highlight ongoing efforts to address these limitations and discuss future directions, including automated LoRA optimization, continual learning, and deployment in ultra-large foundation models. As AI models continue to grow in complexity, LoRA stands out as a scalable and cost-effective solution for fine-tuning, making it an essential tool for researchers and practitioners seeking to adapt LLMs efficiently.

Keywords: low-rank adaptation; large language models; parameter-efficient fine-tuning; deep learning; transfer learning

I. Introduction

Large Language Models (LLMs) have transformed the field of natural language processing (NLP), demonstrating unprecedented capabilities in a wide range of applications, including machine translation, text summarization, dialogue systems, and code generation [1]. Models such as OpenAI's GPT series, Google's T5, Meta's LLaMA, and various other transformer-based architectures have set new benchmarks across numerous NLP tasks. The success of these models is largely attributed to their massive scale, often consisting of billions or even trillions of parameters, trained on vast amounts of text data [2]. However, this rapid advancement has also introduced significant challenges, particularly concerning the fine-tuning and adaptation of such models for specific downstream tasks. Traditional full fine-tuning approaches require updating all model parameters for each new task, leading to high computational costs, substantial memory usage, and long training times [3]. These requirements make fine-tuning infeasible for many researchers and organizations, especially those with limited access to high-performance computing resources [4]. Moreover, as LLMs continue to grow in size, the challenges associated with model adaptation become even more pronounced. This has led to an increasing demand for parameter-efficient fine-tuning methods that enable effective adaptation without the exorbitant computational overhead. One of the most promising techniques to address these challenges is Low-Rank Adaptation (LoRA). LoRA introduces a novel approach to fine-tuning by injecting trainable low-rank matrices into existing weight matrices of LLMs, thereby significantly reducing the number of trainable parameters while maintaining competitive performance [5]. Instead of modifying the full set of model weights, LoRA applies a low-rank decomposition to weight updates, allowing for efficient adaptation without excessive memory or compute requirements [6]. This technique not only reduces the cost of fine-tuning but also facilitates rapid task adaptation, making it particularly useful in scenarios where multiple specialized models are required. The key advantages of LoRA lie in its efficiency, modularity, and versatility [7]. By reducing the trainable parameter count by orders of magnitude compared to full fine-tuning, LoRA enables organizations to fine-tune large models using significantly less GPU memory [8]. This makes it possible to adapt LLMs on consumer-grade hardware, democratizing access to powerful AI models [9]. Additionally, since LoRA operates by introducing small modifications to existing layers rather than overwriting entire model weights, it allows for efficient storage and reusability of fine-tuned adaptations. Multiple task-specific adapters can be stored and switched dynamically, further enhancing the model's flexibility. LoRA has been successfully applied across various domains, including NLP, computer vision, and speech processing [10]. In NLP, it has been used for domain adaptation, sentiment analysis, personalized AI assistants, and knowledge retrieval [11]. Its ability to fine-tune models efficiently while retaining the benefits of large-scale pretraining has made it a crucial tool in both research and industry settings. Moreover, LoRA has been combined with other techniques, such as Prompt Tuning and Prefix Tuning, to further enhance its effectiveness [12]. Recent studies have explored hybrid approaches, demonstrating how LoRA can be integrated with other parameter-efficient tuning (PET) methods to balance efficiency and performance [13]. Despite its advantages, LoRA is not without limitations [14]. One challenge is the trade-off between adaptation efficiency and expressive power. Since LoRA restricts modifications to a low-rank subspace, it may struggle with certain complex tasks that require extensive parameter updates [15]. Additionally, while LoRA reduces memory requirements during training, inference-time efficiency remains an area of active research [16]. Addressing these limitations requires exploring more sophisticated methods of incorporating LoRA into transformer architectures, such as dynamic rank selection, structured pruning, and adaptive tuning strategies [15,17]. This survey provides a comprehensive and structured review of LoRA in the context of LLMs. We begin by discussing the theoretical foundations of LoRA, including its mathematical formulation and underlying principles. We then examine its practical implementations, comparing it with alternative fine-tuning methods and evaluating its impact across various NLP tasks [18]. Furthermore, we highlight recent advancements in LoRA research, including hybrid approaches and optimizations aimed at enhancing its effectiveness. Finally, we explore open challenges and future research directions, such as improving LoRA's applicability to different architectures, optimizing its efficiency for real-time applications, and developing better strategies for balancing performance and computational cost [19]. By consolidating the latest research and insights on LoRA, this survey aims to serve as a valuable resource for researchers, engineers, and practitioners seeking to optimize the adaptation of LLMs. As the demand for efficient fine-tuning methods continues to grow, understanding and leveraging techniques like LoRA will be critical in enabling scalable, accessible, and cost-effective deployment of large-scale AI models [20].

II. Background and Related Work

The rapid advancements in Large Language Models (LLMs) have necessitated the development of efficient fine-tuning techniques to adapt these models to various downstream tasks. Traditional fine-tuning, which involves updating all parameters of a pre-trained model, has proven to be computationally expensive and memory-intensive, especially as model sizes continue to grow [21]. This has led to the exploration of parameter-efficient fine-tuning (PEFT) methods, among which Low-Rank Adaptation (LoRA) has gained significant attention. In this section, we provide an overview of foundational concepts related to LLM fine-tuning, discuss alternative PEFT approaches, and highlight the key developments that have led to the adoption of LoRA [22].

A. Fine-Tuning of Large Language Models

Fine-tuning is the process of adapting a pre-trained LLM to a specific task or domain by further training it on task-specific data [23]. The most common approach, known as full fine-tuning, involves updating all model parameters. While effective, this method has several drawbacks:

- **High computational cost:** Updating billions of parameters requires extensive GPU memory and processing power [24].
- **Storage inefficiency:** Each fine-tuned model requires storing a full copy of the modified weights, making it infeasible to maintain multiple task-specific models [25].
- **Catastrophic forgetting:** Adapting a model to one task may degrade its performance on previously learned tasks if not handled carefully.

These challenges have motivated researchers to explore alternative fine-tuning strategies that are more efficient while preserving the benefits of pre-trained LLMs.

B. Parameter-Efficient Fine-Tuning (PEFT) Approaches

Several parameter-efficient fine-tuning techniques have been proposed to reduce the computational and storage burdens associated with full fine-tuning [26]. Some of the most notable approaches include:

1) Adapter Layers

Adapter layers introduce small, task-specific modules into the transformer architecture while keeping the original model parameters frozen [27]. These lightweight layers are trained while the base model remains unchanged, significantly reducing memory usage [28]. Adapter-based methods allow for efficient multi-task learning, as different adapters can be swapped in and out without modifying the core model. However, they require additional forward-pass computations during inference, which can slightly increase latency.

2) Prompt Tuning and Prefix Tuning

Prompt tuning modifies a model's input rather than its parameters [29]. This approach involves learning a small set of tunable prompt embeddings that guide the model's responses without altering its internal weights [30]. Prefix tuning extends this idea by prepending trainable continuous embeddings to the model's input representations [31]. While effective for certain tasks, these methods often require large amounts of data to match the performance of traditional fine-tuning [32].

3) BitFit

BitFit is an extremely lightweight fine-tuning method that updates only the bias terms of the model's parameters while keeping all other weights frozen. This reduces the number of trainable parameters by several orders of magnitude. Although BitFit works well for some classification tasks, it may struggle with complex generative tasks that require deeper model adaptations.

4) Low-Rank Adaptation (LoRA)

LoRA introduces trainable low-rank matrices into existing weight matrices of an LLM, allowing for efficient adaptation while keeping most of the model's parameters frozen [33]. By restricting updates to a low-rank subspace, LoRA drastically reduces the number of trainable parameters, making it one of the most effective PEFT techniques. It maintains the pre-trained model's knowledge while enabling fast and cost-efficient adaptation [34]. LoRA has been widely adopted due to its balance of efficiency and effectiveness across multiple NLP tasks.

C. Evolution of LoRA and Its Adoption in NLP

The development of LoRA was driven by the need for scalable fine-tuning solutions that mitigate the challenges of large-scale models [35]. LoRA's effectiveness has been demonstrated in various studies, showing that it can achieve performance comparable to full fine-tuning while significantly reducing computational costs [36]. Recent research has explored ways to enhance LoRA further, including hybrid approaches that combine LoRA with other PEFT methods, adaptive rank selection strategies, and optimizations for better inference efficiency. LoRA has been successfully applied in a variety of domains, including:

- **Natural Language Understanding (NLU)**: Tasks such as sentiment analysis, named entity recognition, and text classification benefit from LoRA's ability to fine-tune LLMs efficiently.
- **Text Generation**: LoRA has been integrated into large autoregressive models like GPT to improve domain-specific text generation while maintaining fluency and coherence [37].
- **Multimodal Applications**: Recent work has extended LoRA to multimodal models, enabling efficient adaptation of vision-language models for tasks such as image captioning and visual question answering.

D. Summary

In this section, we provided an overview of the challenges associated with full fine-tuning of LLMs and introduced various parameter-efficient fine-tuning methods. Among these, LoRA has emerged as a leading approach due to its balance between efficiency and performance [38]. In the following sections, we delve deeper into the mathematical foundations of LoRA, its practical implementations, and recent advancements that have further enhanced its effectiveness in adapting LLMs to diverse tasks [39].

III. Mathematical Foundations of LoRA

Low-Rank Adaptation (LoRA) is grounded in the principle of low-rank matrix approximation, which enables efficient fine-tuning of Large Language Models (LLMs) by reducing the number of trainable parameters [40]. This section presents the mathematical formulation of LoRA, detailing its core principles, theoretical justifications, and how it integrates into transformer architectures [41].

A. Low-Rank Decomposition in Neural Networks

In traditional fine-tuning, the weight matrices of a neural network are fully updated during training [42]. However, LoRA assumes that the weight updates during adaptation reside in a low-rank subspace, allowing for a compact representation [43]. Mathematically, let $\mathbf{W} \in \mathbb{R}^{d \times k}$ be a weight matrix in an LLM, where d is the input dimension and k is the output dimension [44]. Instead of updating $\mathbf{W}$ directly, LoRA models the weight update as:

$$\Delta \mathbf{W} = \mathbf{A}\mathbf{B}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{d \times r}$ and $\mathbf{B} \in \mathbb{R}^{r \times k}$ are the low-rank matrices, and $r \ll \min(d, k)$ is the rank of the decomposition [45]. The base model parameters remain frozen while only $\mathbf{A}$ and $\mathbf{B}$ are trained, leading to a significant reduction in the number of trainable parameters [46].

B. Parameter Efficiency and Complexity Reduction

The total number of trainable parameters in standard fine-tuning is $O(dk)$. With LoRA, the number of trainable parameters reduces to:

$$O(dr + rk) = O(r(d + k))[47]. \tag{2}$$

Since r is much smaller than d and k, this leads to a substantial reduction in computational cost. For example, if $r = 8$ in a model with millions of parameters per layer, the storage and training efficiency improve dramatically without significantly impacting performance [48].

C. Integration with Transformer Architectures

LoRA is typically applied to key layers in transformer architectures, such as the self-attention mechanism [49]. In a standard transformer, the attention mechanism computes the output as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}, \tag{3}$$

where $\mathbf{Q} = \mathbf{X}\mathbf{W}_\mathbf{Q}$, $\mathbf{K} = \mathbf{X}\mathbf{W}_\mathbf{K}$, and $\mathbf{V} = \mathbf{X}\mathbf{W}_\mathbf{V}$ are the query, key, and value projections, respectively. In LoRA, the weight matrices $\mathbf{W}_\mathbf{Q}$ and $\mathbf{W}_\mathbf{V}$ are modified as:

$$\mathbf{W}_\mathbf{Q}{}' = \mathbf{W}_\mathbf{Q} + \mathbf{A}_\mathbf{Q}\mathbf{B}_\mathbf{Q}, \quad \mathbf{W}_\mathbf{V}{}' = \mathbf{W}_\mathbf{V} + \mathbf{A}_\mathbf{V}\mathbf{B}_\mathbf{V}. \tag{4}$$

Since the base weights $\mathbf{W}_\mathbf{Q}$ and $\mathbf{W}_\mathbf{V}$ remain frozen, LoRA introduces minimal computational overhead while allowing for effective task adaptation [50].

D. Rank Selection and Performance Trade-Offs

The choice of the rank r in LoRA is crucial for balancing efficiency and model expressiveness. A higher rank allows the adaptation process to capture more complex transformations but increases the number of trainable parameters [51]. Empirical studies suggest that even low-rank settings ($r = 4$ or $r = 8$) can achieve performance comparable to full fine-tuning in many NLP tasks [52]. LoRA's effectiveness can be further understood through singular value decomposition (SVD) [53]. Given a full-rank weight update matrix $\Delta\mathbf{W}$, its optimal low-rank approximation is obtained by truncating its singular value decomposition:

$$\Delta\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \approx \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^T [54]. \tag{5}$$

This insight highlights that LoRA captures the most significant directions of variation while discarding less critical components [55].

E. Comparison with Other Fine-Tuning Methods

LoRA provides several advantages over other fine-tuning approaches:

- **Storage efficiency:** Since only the low-rank matrices are stored, multiple task-specific adaptations can be maintained without redundant full model copies.
- **Reduced computational cost:** Training requires fewer parameters to be updated, leading to faster convergence and lower memory consumption [56].
- **Preservation of pre-trained knowledge:** By keeping the original model weights frozen, LoRA avoids catastrophic forgetting and enables easy model reversibility [57].

F. Summary

LoRA leverages low-rank matrix decomposition to achieve efficient fine-tuning of LLMs while maintaining competitive performance [58]. Its ability to integrate seamlessly with transformer-based architectures, combined with its storage and computational benefits, makes it a powerful tool for scalable adaptation of large models [59]. In the next section, we will explore practical implementations of LoRA, discussing real-world applications, optimization techniques, and empirical performance evaluations [60].

IV. Practical Implementation of LoRA

While the mathematical foundations of Low-Rank Adaptation (LoRA) provide an efficient framework for fine-tuning Large Language Models (LLMs), its practical implementation requires careful integration into training pipelines, optimization strategies, and real-world deployment scenarios [61]. In this section, we discuss the practical aspects of implementing LoRA, including its integration with existing deep learning frameworks, training strategies, evaluation methodologies, and applications in various domains.

A. Integrating LoRA into Deep Learning Frameworks

LoRA has been widely adopted in popular deep learning libraries, making it accessible for researchers and practitioners. Several frameworks provide built-in support for LoRA, including:

- **Hugging Face Transformers:** The Hugging Face library provides APIs to integrate LoRA with models such as GPT, BERT, and T5, enabling efficient fine-tuning with minimal modifications [62].
- **PyTorch LoRA Implementations:** Several PyTorch-based implementations, such as `peft` (Parameter Efficient Fine-Tuning), provide easy-to-use modules for applying LoRA to transformer layers.
- **TensorFlow and JAX Support:** Although less common, LoRA implementations exist for TensorFlow and JAX, allowing for efficient adaptation of LLMs within these ecosystems.

To implement LoRA in practice, developers typically modify transformer layers by introducing low-rank matrices into key projection layers [63]. A simple PyTorch implementation involves replacing standard linear layers with LoRA-adapted layers:

```
import torch
import torch.nn as nn

class LoRALinear(nn.Module):
    def __init__(self, in_features, out_features, rank=8):
        super().__init__()
        self.base_layer = nn.Linear(in_features, out_features, bias=False)
        self.A = nn.Linear(in_features, rank, bias=False)
        self.B = nn.Linear(rank, out_features, bias=False)
        self.B.weight.data.zero_()  # Initialize B to avoid large initial updates

    def forward(self, x):
        return self.base_layer(x) + self.B(self.A(x))
```

This implementation showcases how LoRA modifies a standard linear layer while keeping the original weight matrix frozen [64].

B. Training Strategies for LoRA

Training an LLM with LoRA requires optimizing the low-rank matrices while keeping the original model weights unchanged [65]. The following strategies enhance the effectiveness of LoRA training:

1) Optimizing the Learning Rate

Since LoRA significantly reduces the number of trainable parameters, standard learning rates used for full fine-tuning may not be optimal [66]. Lower learning rates often lead to more stable convergence, while adaptive learning rate schedules (e.g., cosine annealing or warm-up schedules) improve fine-tuning efficiency [67].

2) Gradient Accumulation and Mixed Precision Training

To further optimize training, LoRA can be combined with:

- **Gradient accumulation:** Reduces memory usage by updating gradients over multiple mini-batches [68].
- **Mixed precision training:** Uses lower precision (e.g., FP16 or BF16) for faster computation and reduced memory consumption [69].

3) Task-Specific Adaptation

LoRA is highly effective for domain adaptation and task specialization. Instead of training a separate model for each task, LoRA allows multiple task-specific adapters to be stored and swapped dynamically [70]. For example, in multi-task learning scenarios, different low-rank matrices can be loaded on demand without requiring full retraining [71].

C. Evaluation and Benchmarking

Assessing the effectiveness of LoRA requires rigorous benchmarking against other fine-tuning methods [72]. Common evaluation metrics include:

- **Perplexity (PPL):** Measures how well the fine-tuned model predicts test data, commonly used in language modeling tasks [73].
- **Accuracy and F1-score:** Standard metrics for classification tasks, such as sentiment analysis or named entity recognition [74].
- **BLEU and ROUGE scores:** Used for text generation and summarization tasks to evaluate output quality [75].
- **Computational efficiency:** GPU memory usage, training speed, and inference latency are key factors in evaluating LoRA's efficiency [76].

D. Real-World Applications of LoRA

LoRA has been successfully applied in various domains, demonstrating its versatility and efficiency [77]. Notable applications include:

1) Natural Language Processing (NLP)

- **Chatbots and Virtual Assistants:** LoRA enables fast adaptation of conversational AI models to specific industries (e.g., healthcare, customer service) [78].
- **Machine Translation:** By fine-tuning pre-trained models like mBART, LoRA improves translation quality without excessive computational costs [79].
- **Legal and Financial Text Processing:** LoRA has been used to adapt LLMs for specialized jargon-heavy domains, such as legal document summarization [80].

2) Computer Vision

Recent research has extended LoRA to vision-language models (e.g., CLIP, BLIP), allowing efficient fine-tuning of models for image captioning and multimodal tasks [81].

3) Biomedical and Healthcare Applications

- **Medical Text Analysis:** LoRA has been used to fine-tune BERT-based models for tasks such as clinical report generation and medical coding [82].
- **Drug Discovery:** AI-driven molecular property prediction models benefit from LoRA's efficiency in adapting transformer-based architectures [83].

4) Code Generation and Programming Assistance

LoRA has been applied to fine-tune models like CodeT5 and StarCoder, enhancing their ability to generate code, provide bug fixes, and assist developers in specialized programming languages [84].

E. Challenges and Best Practices

While LoRA offers significant advantages, its implementation is not without challenges [85]. Key considerations include:

- **Rank Selection:** Choosing an appropriate rank r is crucial for maintaining a balance between efficiency and expressiveness.

- **Memory Efficiency:** While LoRA reduces training costs, inference efficiency remains an area of active research [86].
- **Hybrid Fine-Tuning Approaches:** Combining LoRA with other techniques, such as prompt tuning and adapter layers, can further improve performance [87].

F. Summary

LoRA has emerged as a powerful and practical approach for fine-tuning LLMs efficiently. Its seamless integration into popular deep learning frameworks, coupled with its reduced computational footprint, makes it an ideal solution for a wide range of applications [88]. The next section explores recent advancements and ongoing research aimed at further enhancing LoRA's effectiveness and expanding its use cases [89].

V. Recent Advancements and Ongoing Research

The success of Low-Rank Adaptation (LoRA) has sparked extensive research into further improving its efficiency, applicability, and performance [90]. While LoRA has already demonstrated significant advantages in fine-tuning large language models (LLMs), ongoing studies continue to explore new optimizations, hybrid approaches, and theoretical enhancements [91]. This section discusses recent advancements in LoRA-based fine-tuning, including extensions of LoRA, its combination with other parameter-efficient tuning (PET) methods, and novel applications beyond traditional NLP tasks [92].

A. Hybrid Approaches: Combining LoRA with Other Fine-Tuning Techniques

While LoRA significantly reduces the number of trainable parameters, researchers have explored hybrid approaches that combine LoRA with other fine-tuning techniques to maximize efficiency and flexibility [93]. Some notable hybrid methods include:

1) LoRA + Prompt Tuning

Prompt tuning involves learning small continuous embeddings that modify the model's input rather than its parameters [94]. Recent work has combined LoRA with prompt tuning to further reduce the adaptation footprint while maintaining competitive performance [95]. This approach is particularly useful in scenarios where fast task switching is required, such as multi-domain chatbots.

2) LoRA + Prefix Tuning

Prefix tuning extends prompt tuning by introducing learnable embeddings into the model's intermediate representations rather than just the input layer [96]. When used alongside LoRA, this method allows for a balance between expressiveness and computational efficiency, leading to improved results in generative tasks such as machine translation and text summarization.

3) LoRA + Adapter Layers

Adapter layers are small trainable modules inserted within transformer blocks [97]. By combining LoRA with adapter layers, researchers have achieved enhanced model adaptability with minimal memory overhead. This hybrid technique is particularly useful in multilingual NLP, where different adapters can be used for different languages while LoRA fine-tunes shared knowledge [98].

B. Adaptive Rank Selection and Dynamic LoRA

The effectiveness of LoRA is closely tied to the choice of rank r. Traditionally, LoRA uses a fixed rank across all model layers [99]. However, recent research has introduced **adaptive rank selection**, which dynamically adjusts the rank based on layer importance and task complexity. Some key developments in this area include:

- **Layer-wise Rank Allocation:** Instead of assigning a uniform rank to all transformer layers, models can be optimized by using higher ranks in critical layers (e.g., deeper attention layers) and lower ranks in less important layers [100].

- **Task-Specific Rank Optimization:** Algorithms such as evolutionary search or reinforcement learning can be employed to find optimal rank configurations for different tasks [101].
- **Sparse LoRA:** Some studies propose sparsifying the low-rank matrices to further reduce computational requirements while preserving model accuracy [102].

C. LoRA for Multimodal and Cross-Domain Applications

While LoRA has primarily been used in NLP tasks, recent work has explored its application in multimodal learning and cross-domain adaptation [103]. Notable advancements include:

1) LoRA for Vision-Language Models

Vision-language models (VLMs) such as CLIP, BLIP, and Flamingo have demonstrated strong zero-shot learning capabilities [104]. However, fine-tuning these models for domain-specific tasks remains challenging due to their size. LoRA has been successfully integrated into VLMs to enable efficient adaptation for applications such as:

- Image captioning with domain-specific knowledge [105].
- Video understanding for automated content analysis [106].
- Visual question answering (VQA) in specialized fields like medical imaging [107].

2) LoRA for Speech and Audio Processing

Recent studies have explored LoRA's effectiveness in fine-tuning speech recognition and audio generation models. By integrating LoRA into transformer-based architectures such as Whisper or Wav2Vec, researchers have achieved low-cost adaptation for tasks like:

- Domain-specific speech recognition (e.g., medical or legal transcription) [108].
- Emotion-aware conversational AI [109].
- Personalized text-to-speech (TTS) systems.

3) LoRA for Reinforcement Learning and Robotics

Beyond NLP and multimodal applications, LoRA has been investigated in reinforcement learning (RL) settings. Recent work has demonstrated that LoRA can be applied to fine-tune policies in transformer-based RL agents, enabling:

- More efficient policy adaptation in large-scale RL environments [110].
- Parameter-efficient tuning of foundation models for robotics.
- Domain-specific adaptation for embodied AI systems [111].

D. Optimizing LoRA for Efficient Inference

While LoRA significantly reduces training costs, its impact on inference efficiency remains an active area of research [112]. Some recent optimizations aimed at improving inference include:

1) Quantized LoRA

Quantization reduces the precision of model weights to lower-bit representations (e.g., INT8 or FP16) to decrease memory usage and speed up inference [113]. Researchers have explored quantized versions of LoRA to make it even more lightweight, particularly for edge AI applications [114].

2) Fusion of LoRA Adapters

In cases where multiple LoRA adapters are trained for different tasks, researchers have explored methods to merge these adapters into a single model without requiring multiple forward passes [115]. This is particularly useful in multi-task learning scenarios where the model must handle diverse inputs efficiently.

3) LoRA for On-Device AI

Recent work has explored LoRA's role in enabling on-device fine-tuning of language models for personalized AI assistants. By leveraging LoRA's low-memory footprint, models can be fine-tuned on consumer hardware such as smartphones and IoT devices[53,116,117].

E. Theoretical Insights into LoRA's Effectiveness

Several studies have attempted to provide deeper theoretical justifications for why LoRA works so well [118]. Key findings include:

- **LoRA and Model Overparameterization:** Research suggests that large language models contain redundant parameters, making them well-suited for low-rank adaptations [119].
- **Information Flow in LoRA-Modified Networks:** Studies analyzing LoRA-modified transformers indicate that low-rank updates primarily affect key subspaces responsible for task-specific information [120].
- **Optimization Landscapes with LoRA:** Some researchers have analyzed LoRA's impact on the optimization landscape, showing that it enables more stable convergence compared to full fine-tuning [121].

F. Challenges and Future Directions

Despite its many advantages, LoRA has several limitations that warrant further investigation [122]. Key challenges and potential research directions include:

- **LoRA for Highly Specialized Tasks:** While LoRA works well for many tasks, certain applications requiring extensive parameter updates may benefit from hybrid approaches [123].
- **Reducing Inference Overhead:** Although LoRA is efficient during training, methods to optimize inference without introducing additional latency remain an open research question [124].
- **Automated LoRA Configuration:** Developing algorithms that automatically determine the optimal rank and layer placement for LoRA in different architectures can further enhance its usability [125].
- **Expanding LoRA Beyond Transformers:** Most research has focused on transformers, but exploring LoRA's applicability to other architectures, such as CNNs and RNNs, could broaden its impact.

G. Summary

LoRA continues to evolve as a leading parameter-efficient fine-tuning technique [126]. Recent advancements have expanded its capabilities through hybrid approaches, adaptive rank selection, and novel applications beyond NLP [127]. Ongoing research into inference optimization, theoretical foundations, and cross-domain adaptation will further enhance LoRA's role in efficient deep learning. In the next section, we conclude with a discussion on LoRA's long-term impact and potential future developments [128].

VI. Conclusion and Future Perspectives

Low-Rank Adaptation (LoRA) has emerged as a transformative approach to fine-tuning Large Language Models (LLMs) with significantly reduced computational cost and memory requirements [129]. By leveraging low-rank matrix decompositions, LoRA enables efficient adaptation of large-scale pre-trained models without the need to update or store the entire set of parameters [130]. This survey has explored the theoretical foundations, practical implementations, recent advancements, and ongoing research related to LoRA, highlighting its effectiveness across various domains, including natural language processing, vision-language tasks, speech processing, and reinforcement learning.

A. Key Takeaways

The following are the key insights derived from this survey:

- **Parameter Efficiency:** LoRA drastically reduces the number of trainable parameters by decomposing weight updates into low-rank matrices, making fine-tuning feasible for large-scale models.
- **Computational and Memory Benefits:** By keeping the original model weights frozen, LoRA significantly lowers the GPU memory footprint and accelerates training compared to full fine-tuning [131].
- **Seamless Integration:** LoRA has been successfully integrated into widely-used deep learning frameworks such as PyTorch and Hugging Face Transformers, facilitating its adoption by the research and industry communities.
- **Hybrid and Adaptive Techniques:** Recent advancements, such as combining LoRA with prompt tuning, adapter layers, and dynamic rank selection, have further improved its flexibility and effectiveness [132].
- **Multimodal and Cross-Domain Applications:** LoRA has extended beyond NLP and is now being explored in vision-language models, speech processing, and even reinforcement learning.
- **Inference-Time Considerations:** While LoRA optimizes training efficiency, reducing inference overhead remains an important area of research.

B. Future Directions

Despite its impressive benefits, LoRA still presents several open challenges that warrant further exploration [133]. Some promising directions for future research include:

1) Automated LoRA Optimization

Choosing the optimal rank and identifying the most suitable layers for LoRA adaptation remains a manual process in most implementations. Future research could focus on automated methods for rank selection, possibly using reinforcement learning or neural architecture search techniques [134].

2) Reducing Inference Overhead

While LoRA significantly improves training efficiency, it introduces additional computations at inference time due to the added low-rank matrices [135]. Efficient inference techniques, such as matrix fusion or adaptive LoRA integration, could help mitigate this issue [136].

3) Expanding LoRA Beyond Transformers

Most LoRA implementations are tailored for transformer-based architectures [137]. However, extending LoRA to convolutional neural networks (CNNs), recurrent neural networks (RNNs), and graph neural networks (GNNs) could broaden its applicability to new domains such as computer vision and structured data processing [138].

4) Continual Learning and On-Device Adaptation

LoRA's lightweight nature makes it well-suited for continual learning and edge AI applications. Future research could explore how LoRA can enable personalized AI assistants, federated learning scenarios, and on-device fine-tuning with limited computational resources [139].

5) LoRA for Foundation Models

As foundation models continue to grow in scale, LoRA could play a crucial role in enabling efficient adaptation without requiring massive computational resources [140]. Future work could investigate how LoRA can be optimized for ultra-large models like GPT-4, PaLM, and Gemini.

C. Final Thoughts

LoRA represents a paradigm shift in the fine-tuning of large models, offering a scalable and efficient alternative to traditional full-parameter adaptation. As AI models continue to expand in size and complexity, parameter-efficient techniques like LoRA will become increasingly crucial for democratizing access to powerful language models. With ongoing research addressing its limitations and expanding its applications, LoRA is poised to remain a cornerstone of efficient model adaptation in the AI landscape.

References

1. T. Konstantinidis, G. Iacovides, M. Xu, T. G. Constantinides, and D. P. Mandic, "Finllama: Financial sentiment classification for algorithmic trading applications," *arXiv preprint arXiv:2403.12285*, 2024.
2. Y. Zhu, N. Wichers, C. Lin, X. Wang, T. Chen, L. Shu, H. Lu, C. Liu, L. Luo, J. Chen, and L. Meng, "Sira: Sparse mixture of low rank adaptation," *arXiv preprint arXiv.2311.09179*, 2023.
3. T. Chen, T. Ding, B. Yadav, I. Zharkov, and L. Liang, "Lorashear: Efficient large language model structured pruning and knowledge recovery," *arXiv preprint arXiv:2310.18356*, 2023.
4. Y. Chen, S. Qian, H. Tang, X. Lai, Z. Liu, S. Han, and J. Jia, "Longlora: Efficient fine-tuning of long-context large language models," *arXiv preprint arXiv.2309.12307*, 2023.
5. H. Zhang, "Sinklora: Enhanced efficiency and chat capabilities for long-context large language models," *arXiv preprint arXiv.2406.05678*, 2023.
6. J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, "Towards a unified view of parameter-efficient transfer learning," in *International Conference on Learning Representations*, 2022.
7. X. Meng, D. Dai, W. Luo, Z. Yang, S. Wu, X. Wang, P. Wang, Q. Dong, L. Chen, and Z. Sui, "Periodiclora: Breaking the low-rank bottleneck in lora optimization," *arXiv preprint arXiv.2402.16141*, 2024.
8. H. Wang, Z. Xiao, Y. Li, S. Wang, G. Chen, and Y. Chen, "Milora: Harnessing minor singular components for parameter-efficient llm finetuning," *arXiv preprint arXiv:2406.09044*, 2024.
9. F. Zhang and M. Pilanci, "Riemannian preconditioned lora for fine-tuning foundation models," *arXiv preprint arXiv:2402.02347*, 2024.
10. C. Gao, K. Chen, J. Rao, B. Sun, R. Liu, D. Peng, Y. Zhang, X. Guo, J. Yang, and V. S. Subrahmanian, "Higher layers need more lora experts," *arXiv preprint arXiv.2402.08562*, 2024.
11. Y. Gong, Z. Zhan, Q. Jin, Y. Li, Y. Idelbayev, X. Liu, A. Zharkov, K. Aberman, S. Tulyakov, Y. Wang, and J. Ren, "E^2gan: Efficient training of efficient gans for image-to-image translation," *arXiv preprint arXiv:2401.06127*, 2024.
12. H. Qin, X. Ma, X. Zheng, X. Li, Y. Zhang, S. Liu, J. Luo, X. Liu, and M. Magno, "Accurate lora-finetuning quantization of llms via information retention," *arXiv preprint arXiv:2402.05445*, 2024.
13. P. Yadav, L. Choshen, C. Raffel, and M. Bansal, "Compeft: Compression for communicating parameter efficient updates via sparsification and quantization," *arXiv preprint arXiv.2311.13171*, 2023.
14. N. Asadi, M. Beitollahi, Y. H. Khalil, Y. Li, G. Zhang, and X. Chen, "Does combining parameter-efficient modules improve few-shot transfer accuracy?" *arXiv preprint arXiv.2402.15414*, 2024.
15. M. Zhang, H. Chen, C. Shen, Z. Yang, L. Ou, X. Yu, and B. Zhuang, "Loraprune: Pruning meets low-rank parameter-efficient fine-tuning," *arXiv preprint arXiv:2305.18403*, 2023.
16. D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," *arXiv preprint arXiv:2009.03300*, 2020.
17. Y. Zniyed, T. P. Nguyen *et al.*, "Efficient tensor decomposition-based filter pruning," *Neural Networks*, vol. 178, p. 106393, 2024.
18. Z. Liu, J. Lyn, W. Zhu, X. Tian, and Y. Graham, "Alora: Allocating low-rank adaptation for fine-tuning large language models," *arXiv preprint arXiv.2403.16187*, 2024.
19. Y. Xu, L. Xie, X. Gu, X. Chen, H. Chang, H. Zhang, Z. Chen, X. Zhang, and Q. Tian, "Qa-lora: Quantization-aware low-rank adaptation of large language models," *arXiv preprint arXiv:2309.14717*, 2023.
20. Y. Zhang, M. Wang, Y. Wu, P. Tiwari, Q. Li, B. Wang, and J. Qin, "Dialoguellm: Context and emotion knowledge-tuned large language models for emotion recognition in conversations," *arXiv preprint arXiv:2310.11374*, 2024.
21. H. Wang, X. Xiang, Y. Fan, and J. Xue, "Customizing 360-degree panoramas through text-to-image diffusion models," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 4933–4943.
22. F. Zhang, L. Li, J. Chen, Z. Jiang, B. Wang, and Y. Qian, "Increlora: Incremental parameter allocation method for parameter-efficient fine-tuning," *arXiv preprint arXiv.2308.12043*, 2023.
23. Y. Hu, Y. Xie, T. Wang, M. Chen, and Z. Pan, "Structure-aware low-rank adaptation for parameter-efficient fine-tuning," *Mathematics*, vol. 11, no. 20, p. 4317, 2023.

24. Y. Ma, Y. Fan, J. Ji, H. Wang, X. Sun, G. Jiang, A. Shu, and R. Ji, "X-dreamer: Creating high-quality 3d content by bridging the domain gap between text-to-2d and text-to-3d generation," *arXiv preprint arXiv:2312.00085*, 2023.
25. X. He, C. Li, P. Zhang, J. Yang, and X. E. Wang, "Parameter-efficient model adaptation for vision transformers," in *Thirty-Seventh AAAI Conference on Artificial Intelligence*, 2023, pp. 817–825.
26. J. Belofsky, "Token-level adaptation of lora adapters for downstream task generalization," in *6th Artificial Intelligence and Cloud Computing Conference*, 2023, pp. 168–172.
27. K. Suri, P. Mishra, S. Saha, and A. Singh, "Suryakiran at mediqa-sum 2023: Leveraging lora for clinical dialogue summarization," in *Working Notes of the Conference and Labs of the Evaluation Forum*, 2023, pp. 1720–1735.
28. S. Li, H. Lu, T. Wu, M. Yu, Q. Weng, X. Chen, Y. Shan, B. Yuan, and W. Wang, "Caraserve: Cpu-assisted and rank-aware lora serving for generative llm inference," *arXiv preprint arXiv:2401.11240*, 2024.
29. S. Li, "Diffstyler: Diffusion-based localized image style transfer," *arXiv preprint arXiv:2403.18461*, 2024.
30. R. Miles, P. Reddy, I. Elezi, and J. Deng, "Velora: Memory efficient training using rank-1 sub-token projections," *arXiv preprint arXiv:2405.17991*, 2024.
31. R. Pan, X. Liu, S. Diao, R. Pi, J. Zhang, C. Han, and T. Zhang, "LISA: layerwise importance sampling for memory-efficient large language model fine-tuning," *arXiv preprint arXiv.2403.17919*, 2024.
32. M. Frank, P. Wolfe *et al.*, "An algorithm for quadratic programming," *Naval research logistics quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.
33. A. Wang, M. Islam, M. Xu, Y. Zhang, and H. Ren, "SAM meets robotic surgery: An empirical study on generalization, robustness and adaptation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2023, pp. 234–244.
34. A. P. Gema, L. Daines, P. Minervini, and B. Alex, "Parameter-efficient fine-tuning of llama for the clinical domain," *arXiv preprint arXiv:2307.03042*, 2023.
35. Y. Sui, M. Yin, Y. Gong, J. Xiao, H. Phan, and B. Yuan, "ELRT: efficient low-rank training for compact convolutional neural networks," *arXiv preprint arXiv.2401.10341*, 2024.
36. S. Kim, H. Yang, Y. Kim, Y. Hong, and E. Park, "Hydra: Multi-head low-rank adaptation for parameter efficient fine-tuning," *Neural Networks*, p. 106414, 2024.
37. A. Bhatti, S. Parmar, and S. Lee, "SM70: A large language model for medical devices," *arXiv preprint arXiv:2312.06974*, 2023.
38. Y. Sun, Z. Li, Y. Li, and B. Ding, "Improving LoRA in privacy-preserving federated learning," *arXiv preprint arXiv:2403.12313*, 2024.
39. Y. Liu, C. An, and X. Qiu, "Y-tuning: An efficient tuning paradigm for large-scale pre-trained models via label representation learning," *Frontiers of Computer Science*, vol. 18, no. 4, p. 184320, 2024.
40. Y. Li, Y. Yu, C. Liang, P. He, N. Karampatziakis, W. Chen, and T. Zhao, "Loftq: Lora-fine-tuning-aware quantization for large language models," *arXiv preprint arXiv:2310.08659*, 2023.
41. J. S. Smith, P. Cascante-Bonilla, A. Arbelle, D. Kim, R. Panda, D. D. Cox, D. Yang, Z. Kira, R. Feris, and L. Karlinsky, "Construct-vl: Data-free continual structured VL concepts learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 994–15 004.
42. A. Han, J. Li, W. Huang, M. Hong, A. Takeda, P. Jawanpuria, and B. Mishra, "Sltrain: a sparse plus low-rank approach for parameter and memory efficient pretraining," *arXiv preprint arXiv:2406.02214*, 2024.
43. S. Ayupov and N. Chirkova, "Parameter-efficient finetuning of transformers for source code," *arXiv preprint arXiv:2212.05901*, 2022.
44. T. Huang, Y. Zeng, Z. Zhang, W. Xu, H. Xu, S. Xu, R. W. H. Lau, and W. Zuo, "Dreamcontrol: Control-based text-to-3d generation with 3d self-prior," *arXiv preprint arXiv:2312.06439*, 2023.
45. A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts *et al.*, "Stable video diffusion: Scaling latent video diffusion models to large datasets," *arXiv preprint arXiv:2311.15127*, 2023.
46. S. Liu, J. Keung, Z. Yang, F. Liu, Q. Zhou, and Y. Liao, "Delving into parameter-efficient fine-tuning in code change learning: An empirical study," *arXiv preprint arXiv:2402.06247*, 2024.
47. R. Zhang, R. Qiang, S. A. Somayajula, and P. Xie, "Autolora: Automatically tuning matrix ranks in low-rank adaptation based on meta learning," *arXiv preprint arXiv.2403.09113*, 2024.
48. T. Bornheim, N. Grieger, P. G. Blaneck, and S. Bialonski, "Speaker attribution in german parliamentary debates with qlora-adapted large language models," *arXiv preprint arXiv:2309.09902*, 2024.

49. J. H. Yeo, S. Han, M. Kim, and Y. M. Ro, "Where visual speech meets language: VSP-LLM framework for efficient and context-aware visual speech processing," *arXiv preprint arXiv:2402.15151*, 2024.
50. R. Qiang, R. Zhang, and P. Xie, "Bilora: A bi-level optimization framework for overfitting-resilient low-rank adaptation of large pre-trained models," *arXiv preprint arXiv:2403.13037*, 2024.
51. J. Gallego-Posada, J. Ramirez, A. Erraqabi, Y. Bengio, and S. Lacoste-Julien, "Controlled sparsity via constrained optimization or: How I learned to stop tuning penalties and love constraints," in *Annual Conference on Neural Information Processing Systems*, 2022.
52. Y. Zhai, H. Zhang, Y. Lei, Y. Yu, K. Xu, D. Feng, B. Ding, and H. Wang, "Uncertainty-penalized reinforcement learning from human feedback with diverse reward lora ensembles," *arXiv preprint arXiv:2401.00243*, 2024.
53. F. Jin, Y. Liu, and Y. Tan, "Derivative-free optimization for low-rank adaptation in large language models," *arXiv preprint arXiv:2403.01754*, 2024.
54. U. Jang, J. D. Lee, and E. K. Ryu, "Lora training in the NTK regime has no spurious local minima," *arXiv preprint arXiv.2402.11867*, 2024.
55. Y. Shen, Z. Xu, Q. Wang, Y. Cheng, W. Yin, and L. Huang, "Multimodal instruction tuning with conditional mixture of lora," *arXiv preprint arXiv.2402.15896*, 2024.
56. A. N. Lee, C. J. Hunter, and N. Ruiz, "Platypus: Quick, cheap, and powerful refinement of llms," *arXiv preprint arXiv:2308.07317*, 2023.
57. H. Zhou, X. Lu, W. Xu, C. Zhu, and T. Zhao, "Lora-drop: Efficient lora parameter pruning based on output evaluation," *arXiv preprint arXiv:2402.07721*, 2024.
58. B. Zi, X. Qi, L. Wang, J. Wang, K. Wong, and L. Zhang, "Delta-lora: Fine-tuning high-rank parameters with the delta of low-rank matrices," *arXiv preprint arXiv.2309.02411*, 2023.
59. J. Sun, D. Fu, Y. Hu, S. Wang, R. Rassin, D.-C. Juan, D. Alon, C. Herrmann, S. van Steenkiste, R. Krishna *et al.*, "Dreamsync: Aligning text-to-image generation with image understanding feedback," in *Synthetic Data for Computer Vision Workshop@ CVPR 2024*, 2023.
60. S. Luo, Y. Tan, S. Patil, D. Gu, P. von Platen, A. Passos, L. Huang, J. Li, and H. Zhao, "Lcm-lora: A universal stable-diffusion acceleration module," *arXiv preprint arXiv:2311.05556*, 2023.
61. F. Meng, Z. Wang, and M. Zhang, "Pissa: Principal singular values and singular vectors adaptation of large language models," *arXiv preprint arXiv:2404.02948*, 2024.
62. M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous federated learning: State-of-the-art and research challenges," *ACM Computing Surveys*, vol. 56, no. 3, pp. 79:1–79:44, 2024.
63. H. Li, F. Koto, M. Wu, A. F. Aji, and T. Baldwin, "Bactrian-x: Multilingual replicable instruction-following models with low-rank adaptation," *arXiv preprint arXiv:2305.15011*, 2023.
64. M. Valipour, M. Rezagholizadeh, I. Kobyzev, and A. Ghodsi, "Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation," *arXiv preprint arXiv:2210.07558*, 2022.
65. H. Sidahmed, S. Phatale, A. Hutcheson, Z. Lin, Z. Chen, Z. Yu, J. Jin, R. Komarytsia, C. Ahlheim, Y. Zhu, S. Chaudhary, B. Li, S. Ganesh, B. Byrne, J. Hoffmann, H. Mansoor, W. Li, A. Rastogi, and L. Dixon, "Perl:parameter efficient reinforcement learning from human feedback," *arXiv preprint arXiv:2403.10704*, 2024.
66. Y. Sun, M. Li, Y. Cao, K. Wang, W. Wang, X. Zeng, and R. Zhao, "To be or not to be? an exploration of continuously controllable prompt engineering," *arXiv preprint arXiv:2311.09773*, 2023.
67. S. Quan, "Dmoerm: Recipes of mixture-of-experts for effective reward modeling," *arXiv preprint arXiv:2403.01197*, 2024.
68. L. Zhang, L. Zhang, S. Shi, X. Chu, and B. Li, "Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning," *arXiv preprint arXiv:2308.03303*, 2023.
69. X. Wang, L. Aitchison, and M. Rudolph, "Lora ensembles for large language model fine-tuning," *arXiv preprint arXiv.2310.00035*, 2023.
70. J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, 2019, pp. 4171–4186.
71. Y. Zhang, J. Wang, L. Yu, D. Xu, and X. Zhang, "Personalized lora for human-centered text understanding," in *Thirty-Eighth AAAI Conference on Artificial Intelligence*, 2024, pp. 19 588–19 596.
72. E. B. Zaken, Y. Goldberg, and S. Ravfogel, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022, pp. 1–9.

73. Y. Zhu, X. Yang, Y. Wu, and W. Zhang, "Parameter-efficient fine-tuning with layer pruning on free-text sequence-to-sequence modeling," *arXiv preprint arXiv:2305.08285*, 2023.
74. A. X. Yang, M. Robeyns, X. Wang, and L. Aitchison, "Bayesian low-rank adaptation for large language models," *arXiv preprint arXiv.2308.13111*, 2023.
75. L. Chen, Z. Ye, Y. Wu, D. Zhuo, L. Ceze, and A. Krishnamurthy, "Punica: Multi-tenant lora serving," in *Proceedings of Machine Learning and Systems*, 2024, pp. 1–13.
76. H. Ding, J. Gao, Y. Yuan, and Q. Wang, "Samlp: A customized segment anything model for license plate detection," *arXiv preprint arXiv:2401.06374*, 2024.
77. Y. Zeng and K. Lee, "The expressive power of low-rank adaptation," *arXiv preprint arXiv.2310.17513*, 2023.
78. A. Khandelwal, "Infusion: Inject and attention fusion for multi concept zero-shot text-based video editing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3017–3026.
79. C. Louizos, M. Welling, and D. P. Kingma, "Learning sparse neural networks through l_0 regularization," *arXiv preprint arXiv.1712.01312*, 2017.
80. Q. Liu, X. Wu, X. Zhao, Y. Zhu, D. Xu, F. Tian, and Y. Zheng, "Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications," *arXiv preprint arXiv.2310.18339*, 2023.
81. S. Yang, Y. Zhou, Z. Liu, and C. C. Loy, "Rerender A video: Zero-shot text-guided video-to-video translation," in *SIGGRAPH Asia 2023 Conference Papers*, 2023, pp. 1–11.
82. S. Liu, C. Wang, H. Yin, P. Molchanov, Y. F. Wang, K. Cheng, and M. Chen, "Dora: Weight-decomposed low-rank adaptation," *arXiv preprint arXiv.2402.09353*, 2024.
83. N. Ding, X. Lv, Q. Wang, Y. Chen, B. Zhou, Z. Liu, and M. Sun, "Sparse low-rank adaptation of pre-trained language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 4133–4145.
84. V. Fomenko, H. Yu, J. Lee, S. Hsieh, and W. Chen, "A note on lora," *arXiv preprint arXiv:2404.05086*, 2024.
85. S. Zhang, Z. Chen, S. Chen, Y. Shen, Z. Sun, and C. Gan, "Improving reinforcement learning from human feedback with efficient reward model ensemble," *arXiv preprint arXiv:2401.16635*, 2024.
86. J. Zhang, S. Chen, J. Liu, and J. He, "Composing parameter-efficient modules with arithmetic operations," *arXiv preprint arXiv.2306.14870*, 2023.
87. N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C. Chan, W. Chen, J. Yi, W. Zhao, X. Wang, Z. Liu, H. Zheng, J. Chen, Y. Liu, J. Tang, J. Li, and M. Sun, "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nat. Mac. Intell.*, vol. 5, no. 3, pp. 220–235, 2023.
88. P. Ren, C. Shi, S. Wu, M. Zhang, Z. Ren, M. de Rijke, Z. Chen, and J. Pei, "Mini-ensemble low-rank adapters for parameter-efficient fine-tuning," *arXiv preprint arXiv.2402.17263*, 2024.
89. W. Feng, L. Zhu, and L. Yu, "Cheap lunch for medical image segmentation by fine-tuning SAM on few exemplars," *arXiv preprint arXiv:2308.14133*, 2023.
90. H. Yang, Y. Wang, X. Xu, H. Zhang, and Y. Bian, "Can we trust llms? mitigate overconfidence bias in llms through knowledge transfer," *arXiv preprint arXiv.2405.16856*, 2024.
91. E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *The Tenth International Conference on Learning Representations*, 2022.
92. W. Jiang, B. Lin, H. Shi, Y. Zhang, Z. Li, and J. T. Kwok, "Effective and parameter-efficient reusing fine-tuned models," *arXiv preprint arXiv.2310.01886*, 2023.
93. K. Bałazy, M. Banaei, K. Aberer, and J. Tabor, "Lora-xs: Low-rank adaptation with extremely small number of parameters," *arXiv preprint arXiv:2405.17604*, 2024.
94. V. Lialin, S. Muckatira, N. Shivagunde, and A. Rumshisky, "Relora: High-rank training through low-rank updates," in *The Twelfth International Conference on Learning Representations*, 2023.
95. J. Zhao, Z. Zhang, B. Chen, Z. Wang, A. Anandkumar, and Y. Tian, "Galore: Memory-efficient LLM training by gradient low-rank projection," *arXiv preprint arXiv.2403.03507*, 2024.
96. Y. Yan, S. Tang, Z. Shi, and Q. Yang, "FeDeRA: Efficient fine-tuning of language models in federated learning leveraging weight decomposition," *arXiv preprint arXiv:2404.18848*, 2024.
97. W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.
98. T. Liu and B. K. H. Low, "Goat: Fine-tuned llama outperforms GPT-4 on arithmetic tasks," *arXiv preprint arXiv:2305.14201*, 2023.
99. S. Woo, B. Park, B. Kim, M. Jo, S. Kwon, D. Jeon, and D. Lee, "Dropbp: Accelerating fine-tuning of large language models by dropping backward propagation," *arXiv preprint arXiv:2402.17812*, 2024.

doi:10.20944/preprints202502.1637.v1

100. S. Malladi, A. Wettig, D. Yu, D. Chen, and S. Arora, “A kernel-based view of language model fine-tuning,” in *International Conference on Machine Learning*, 2023, pp. 23 610–23 641.
101. K. Yu, J. Liu, M. Feng, M. Cui, and X. Xie, “Boosting3d: High-fidelity image-to-3d by boosting 2d diffusion prior to 3d prior with progressive learning,” *arXiv preprint arXiv:2311.13617*, 2023.
102. R. Chitale, A. Vaidya, A. Kane, and A. Ghotkar, “Task arithmetic with lora for continual learning,” *arXiv preprint arXiv.2311.02428*, 2023.
103. J. Yang, “Longqlora: Efficient and effective method to extend context length of large language models,” *arXiv preprint arXiv:2311.04879*, 2023.
104. Z. Zhao, L. Gan, G. Wang, W. Zhou, H. Yang, K. Kuang, and F. Wu, “Loraretriever: Input-aware lora retrieval and composition for mixed tasks in the wild,” *arXiv preprint arXiv.2402.09997*, 2024.
105. A. Toma, P. R. Lawler, J. Ba, R. G. Krishnan, B. B. Rubin, and B. Wang, “Clinical camel: An open-source expert-level medical language model with dialogue-based knowledge encoding,” *arXiv preprint arXiv:2305.12031*, 2023.
106. R. Roberson, G. Kaki, and A. Trivedi, “Analyzing the effectiveness of large language models on text-to-sql synthesis,” *arXiv preprint arXiv:2401.12379*, 2024.
107. Z. Chen, H. Huang, A. Andrusenko, O. Hrinchuk, K. C. Puvvada, J. Li, S. Ghosh, J. Balam, and B. Ginsburg, “SALM: speech-augmented language model with in-context learning for speech recognition and translation,” *arXiv preprint arXiv:2310.09424*, 2023.
108. L. Yi, H. Yu, G. Wang, X. Liu, and X. Li, “pFedLoRA: Model-Heterogeneous Personalized Federated Learning with LoRA Tuning,” *arXiv preprint arXiv:2310.13283*, 2023.
109. Y. Guo, C. Yang, A. Rao, Y. Wang, Y. Qiao, D. Lin, and B. Dai, “Animatediff: Animate your personalized text-to-image diffusion models without specific tuning,” *arXiv preprint arXiv:2307.04725*, 2023.
110. B. Geshkovski, C. Letrouit, Y. Polyanskiy, and P. Rigollet, “The emergence of clusters in self-attention dynamics,” in *Annual Conference on Neural Information Processing Systems*, 2023.
111. S. Sun, D. Gupta, and M. Iyyer, “Exploring the impact of low-rank adaptation on the performance, efficiency, and regularization of RLHF,” *arXiv preprint arXiv:2309.09055*, 2023.
112. Y. Wu, Y. Xiang, S. Huo, Y. Gong, and P. Liang, “Lora-sp: streamlined partial parameter adaptation for resource efficient fine-tuning of large language models,” in *Third International Conference on Algorithms, Microchips, and Network Applications*, 2024, pp. 488–496.
113. Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, and Z. Sui, “A survey for in-context learning,” *arXiv preprint arXiv.2301.00234*, 2023.
114. H. Jeon, Y. Kim, and J.-j. Kim, “L4q: Parameter efficient quantization-aware training on large language models via lora-wise lsq,” *arXiv preprint arXiv:2402.04902*, 2024.
115. Y. Deng, R. Wang, Y. Zhang, Y. Tai, and C. Tang, “Dragvideo: Interactive drag-style video editing,” *arXiv preprint arXiv:2312.02216*, 2023.
116. Z. Chen, Z. Wang, Z. Wang, H. Liu, Z. Yin, S. Liu, L. Sheng, W. Ouyang, Y. Qiao, and J. Shao, “Octavius: Mitigating task interference in mllms via moe,” *arXiv preprint arXiv.2311.02684*, 2023.
117. Y. Zniyed, T. P. Nguyen *et al.*, “Enhanced network compression through tensor decompositions and pruning,” *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
118. I. J. Goodfellow, Y. Bengio, and A. C. Courville, *Deep Learning*, ser. Adaptive computation and machine learning. MIT Press, 2016.
119. N. Hansen and A. Ostermeier, “Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation,” in *Proceedings of IEEE international conference on evolutionary computation*, 1996, pp. 312–317.
120. S. Wang, L. Chen, J. Jiang, B. Xue, L. Kong, and C. Wu, “Lora meets dropout under a unified framework,” *arXiv preprint arXiv:2403.00812*, 2024.
121. B. Geshkovski, C. Letrouit, Y. Polyanskiy, and P. Rigollet, “A mathematical perspective on transformers,” *arXiv preprint arXiv.2312.10794*, 2023.
122. M. Zhong, Y. Shen, S. Wang, Y. Lu, Y. Jiao, S. Ouyang, D. Yu, J. Han, and W. Chen, “Multi-lora composition for image generation,” *arXiv preprint arXiv.2402.16843*, 2024.
123. Y. Sheng, S. Cao, D. Li, C. Hooper, N. Lee, S. Yang, C. Chou, B. Zhu, L. Zheng, K. Keutzer *et al.*, “S-lora: Serving thousands of concurrent lora adapters,” *arXiv preprint arXiv:2311.03285*, 2023.
124. J. Li, Y. Lei, Y. Bian, D. Cheng, Z. Ding, and C. Jiang, “Ra-cfgpt: Chinese financial assistant with retrieval-augmented large language model,” *Frontiers of Computer Science*, vol. 18, no. 5, p. 185350, 2024.

125. S. Yoo, K. Kim, V. G. Kim, and M. Sung, "As-plausible-as-possible: Plausibility-aware mesh deformation using 2d diffusion priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4315–4324.
126. W. Tan, W. Zhang, S. Liu, L. Zheng, X. Wang, and B. An, "True knowledge comes from practice: Aligning llms with embodied environments via reinforcement learning," *arXiv preprint arXiv:2401.14151*, 2024.
127. Z. Qi, X. Tan, S. Shi, C. Qu, Y. Xu, and Y. Qi, "PILLOW: enhancing efficient instruction fine-tuning via prompt matching," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2023, pp. 471–482.
128. Y. Gou, Z. Liu, K. Chen, L. Hong, H. Xu, A. Li, D. Yeung, J. T. Kwok, and Y. Zhang, "Mixture of cluster-conditional lora experts for vision-language instruction tuning," *arXiv preprint arXiv.2312.12379*, 2023.
129. A. Aghajanyan, S. Gupta, and L. Zettlemoyer, "Intrinsic dimensionality explains the effectiveness of language model fine-tuning," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021, pp. 7319–7328.
130. D. Biderman, J. J. G. Ortiz, J. Portes, M. Paul, P. Greengard, C. Jennings, D. King, S. Havens, V. Chiley, J. Frankle, C. Blakeney, and J. P. Cunningham, "Lora learns less and forgets less," *arXiv preprint arXiv.2405.09673*, 2024.
131. Y. Ge, Y. Ge, Z. Zeng, X. Wang, and Y. Shan, "Planting a SEED of vision in large language model," *arXiv preprint arXiv:2307.08041*, 2023.
132. D. J. Kopiczko, T. Blankevoort, and Y. M. Asano, "Vera: Vector-based random matrix adaptation," *arXiv preprint arXiv:2310.11454*, 2023.
133. T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances in neural information processing systems*, 2016, p. 901.
134. A. X. Yang, M. Robeyns, T. Coste, J. Wang, H. Bou-Ammar, and L. Aitchison, "Bayesian reward models for LLM alignment," *arXiv preprint arXiv:2402.13210*, 2024.
135. K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi, "Winogrande: An adversarial winograd schema challenge at scale," *Communications of the ACM*, vol. 64, no. 9, pp. 99–106, 2021.
136. Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, Q. Guo, M. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv.2312.10997*, 2023.
137. Z. Ye, L. Lovell, A. Faramarzi, and J. Ninic, "Sam-based instance segmentation models for the automation of structural damage detection," *arXiv preprint arXiv:2401.15266*, 2024.
138. A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.
139. J. Shi and H. Hua, "Space narrative: Generating images and 3d scenes of chinese garden from text using deep learning," in *xArch–creativity in the age of digital reproduction symposium*, 2023, pp. 236–243.
140. B. Liao and C. Monz, "Apiq: Finetuning of 2-bit quantized large language model," *arXiv preprint arXiv:2402.05147*, 2024.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.