LFMA: Parameter-Efficient Fine-Tuning via Layerwise Fourier Masked Adapter with Top-k Frequency Selection

Anonymous Author(s)
Affiliation
Address
email

Abstract

Low-Rank Adaptation (LoRA) has been widely adopted as a Parameter Efficient Fine-Tuning method for large models such as Large Language Models (LLMs) and Vision Transformer (ViT). However, it encounters scalability limitations, particularly in storage and deployment efficiency, when applied to large foundational models or a wide range of task-specific adaptations, due to the overhead of managing multiple adapters and the reliance on linearly constrained spaces for representation. To address these limitations, Fourier Fine-Tuning (FourierFT) has emerged as an alternative, leveraging the Fourier transform to achieve comparable or superior performance to LoRA while utilizing significantly fewer trainable parameters. Nevertheless, FourierFT targets the entire frequency spectrum to apply updates, which may cause inefficiency, particularly when the meaningful information is concentrated within a specific set of frequency components. The magnitude of each Fourier component reflects its contribution to the original weight update. Thus, selecting Top-K components with the highest magnitudes effectively captures the most informative changes. Therefore, we propose Layerwise Fourier Masked Adapter (LFMA), which selectively fine-tunes using Top-K informative frequency components and resulting in an enhancement of both parameter efficiency and task-specific adaptation. Empirically, we showed similar or better performance than FourierFT in four tasks: image classification, instruction tuning, natural language generation, and natural language understanding. These results demonstrate that selectively fine-tuning in the most informative frequency components is able to push the limits of adapter-based fine-tuning further in terms of scalability and expressivity.

1 Introduction

The advent of Large Foundation Models (LFMs) has marked a paradigm shift in artificial intelligence, moving away from training task-specific models from scratch towards a pre-train and fine-tune methodology. These models, such as GPT-3, RoBERTa, and Vision Transformer (ViT), are pre-trained on vast unlabeled datasets, acquiring a broad and generalizable understanding of language or visual data (12; 13; 14). Their impressive zero-shot and few-shot learning capabilities are a testament to the power of scaling, with modern architectures often containing billions or even trillions of parameters (17; 26). However, this immense scale presents a significant challenge for adaptation to specialized downstream tasks.

The conventional method for adaptation, full fine-tuning (FFT), involves updating all of the model's parameters on a task-specific dataset. While often yielding high performance, FFT is exceptionally resource-intensive (15). It creates a "vicious cycle" of operational complexity: for every new task,

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

a complete, multi-gigabyte copy of the model must be stored, managed, and deployed. This not only leads to exorbitant storage costs but also complicates model versioning and serving, creating a significant barrier to the widespread, customized application of LFMs (18). This critical bottleneck has catalyzed the development of more efficient adaptation strategies.

Parameter-Efficient Fine-Tuning (PEFT) methods have emerged as a compelling solution to this challenge, aiming to match the performance of FFT while tuning only a minuscule fraction of the model's parameters (19; 25). These techniques generally follow an additive philosophy, where the large pre-trained weights are frozen, and small, trainable modules are inserted into the model's architecture. Early approaches like Adapters introduced bottleneck-style modules between transformer layers (1). More recently, Low-Rank Adaptation (LoRA) has become the predominant PEFT method (2). LoRA is predicated on the hypothesis that the weight update matrix during adaptation has a low intrinsic rank. It operationalizes this by decomposing the update into two low-rank matrices, dramatically reducing the number of trainable parameters. The elegance and effectiveness of LoRA have made it a cornerstone of modern LFM customization (21; 22; 23; 24).

Despite its success, LoRA's reliance on a low-rank decomposition imposes a linear information bottleneck on the adaptation process. This may not be sufficiently expressive for all tasks, potentially discarding useful higher-rank information. This limitation motivates the search for alternative parameterization spaces that offer richer, non-linear representations. The frequency domain presents a powerful candidate space. FourierFT recently pioneered this direction by using the Discrete Fourier Transform (DFT) to perform fine-tuning directly on the spectral coefficients of the weight updates (4). This shift in domain allows for capturing complex patterns with even greater parameter efficiency than LoRA.

However, existing frequency-domain methods like FourierFT operate under a paradigm of uniform updates, modifying all spectral components equally. This approach overlooks a fundamental principle from signal processing: natural signals—and, by extension, the informational signals within neural network weights—are often sparse in a transformed basis like the Fourier domain. This spectral sparsity implies that the most crucial information for adaptation is concentrated in a small subset of dominant frequency components. Updating the entire spectrum is therefore inefficient and potentially detrimental, as it may introduce noise by altering low-magnitude, uninformative frequencies.

In this work, we embrace the principle of spectral sparsity to push the boundaries of parameter efficiency. We propose the Layerwise Fourier Masked Adapter (LFMA), a novel PEFT method that combines the benefits of frequency-domain adaptation with a targeted, sparsity-aware update mechanism. Instead of a uniform update, LFMA identifies the Top-K most significant frequency components—those with the highest magnitude—and exclusively optimizes this sparse subset (5; 7). By concentrating the model's learning capacity on the most impactful frequencies, LFMA achieves a more precise and efficient adaptation. Our contributions are threefold:

1. We introduce and motivate the concept of spectral sparsity for parameter-efficient fine-tuning.
2. We present LFMA, a novel adapter that operationalizes this concept by selectively tuning Top-K frequency components.
3. We conduct extensive experiments across diverse benchmarks in computer vision and natural language understanding, demonstrating that LFMA achieves competitive or superior performance to state-of-the-art PEFT methods while requiring a similar or often smaller parameter budget.

2 Related Work

2.1 Parameter-Efficient Fine-Tuning

With the growing scale of large models, Parameter-Efficient Fine-Tuning (PEFT) methods have received increasing attention as a way to adapt models to downstream tasks without updating all parameters (19; 25). These methods significantly reduce computational cost and storage requirements while enabling efficient task-specific adaptation.

Adapter (1) One of the earliest PEFT methods, adapters introduce small trainable modules in parallel to the pre-trained weights, allowing the base model to remain frozen. While effective,

Method	CIFAR10	CIFAR100	OxfordPets	RESISC45	StanfordCars	EuroSAT	FGVC	Avg.
LFMA (ViT-L)	96.58	86.42	89.86	93.79	**82.66**	98.67	**78.73**	89.53
LFMA (ViT-B)	**97.78**	88.68	92.59	**95.00**	79.83	**98.74**	77.20	**89.97**
Full FT (ViT-L)	96.62	85.69	94.24	93.02	83.05	98.26	77.08	89.71
LoRA (ViT-L)	96.60	86.98	94.63	91.30	67.40	97.85	51.15	83.70
FourierFT (ViT-L)	96.55	85.48	**94.65**	91.79	73.29	97.88	60.11	85.68
Full FT (ViT-B)	97.32	**89.02**	91.58	94.24	80.33	98.59	75.97	89.58
LoRA (ViT-B)	97.18	88.66	91.51	90.81	45.93	97.95	46.29	79.76
FourierFT (ViT-B)	97.09	88.09	91.49	92.37	56.91	98.32	53.57	82.55

Table 1: Performance comparison on seven image-classification datasets based on accuracy (%). For LFMA, we used following settings; top-k=0.05, $\alpha = 12$.

this approach increases architectural complexity. LoRA (2) improves efficiency by decomposing weight updates into low-rank matrices, tuning only a small number of parameters. It is now widely regarded as the standard approach for PEFT in large-scale models such as LLMs (26). However, it suffers from limitations in scalability, as managing numerous adapters across tasks incurs storage and deployment overhead. Other popular PEFT methods include prompt-tuning and prefix-tuning, which modify the input embeddings or attention mechanisms, respectively (27; 28). Parameter-Efficient Fine-Tuning with Discrete Fourier Transform (4) This approach addresses LoRA's limitations by transforming model weights into the frequency domain using Discrete Fourier Transform (DFT) (6), enabling richer, non-linear parameter updates. It marks a shift toward frequency-based PEFT, offering greater expressive power with fewer trainable parameters.

Our work builds upon this foundation by introducing the Layerwise Fourier Masked Adapter (LFMA), which extends parameter-efficient fine-tuning into the frequency domain. Unlike prior methods such as LoRA or DFT-based tuning that update all parameters or spectral components (2; 4), LFMA selectively tunes only the Top-K frequency components (5; 7) with the highest magnitudes. This selective tuning significantly reduces the number of trainable parameters while maintaining or improving performance across downstream tasks, thus advancing the scalability and practicality of PEFT in large-scale models.

2.2 Frequency Domain Learning

Traditionally, neural networks operate in the spatial or temporal domain. Recently, frequency domain learning (8; 9; 10; 11) has emerged as a promising direction for improving learning efficiency, compression, and representation power by manipulating neural weights or features in the spectral space (29; 30).

Fourier Neural Operator (3) Proposed for solving partial differential equations, the Fourier Neural Operator applies global convolutions in the frequency domain, demonstrating the representational benefits of spectral operations and resolution-invariant learning in continuous settings. Parameter-Efficient Fine-Tuning with Discrete Fourier Transform (4) This method applies the Discrete Fourier Transform (6) to model weights, enabling fine-tuning directly in the frequency domain. By modifying the spectral coefficients instead of the raw weights, the method overcomes linear constraints of LoRA and achieves comparable performance with significantly fewer trainable parameters (2). However, it still updates the full frequency spectrum uniformly, which may include components with negligible informational value.

In contrast to prior approaches that apply uniform updates across the entire frequency spectrum, our proposed Layerwise Fourier Masked Adapter (LFMA) focuses on selectively updating only the Top-K most informative frequency components (5; 7) per layer. These components are identified based on their magnitudes, which indicate their contribution to the original weight update. This spectral masking strategy enhances both representation precision and adaptation efficiency, pushing the boundaries of frequency-based learning in terms of both task-specific expressiveness and computational scalability (31).

3 Methodology

Algorithm 1 PyTorch-style pseudocode for **LayerwiseFGFourierFTAdapter**.

```
class LayerwiseFourierFTAdapter(nn.Module):
    def init(self, alpha, base_layer, delta_W_init, top_k_ratio):
        super().init()
        self.alpha = alpha
        self.base_layer = base_layer
        for p in self.base_layer.parameters():
            p.requires_grad = False

        d1, d2 = delta_W_init.shape
        W_freq = torch.fft.fft2(delta_W_init)
        magnitude = torch.abs(W_freq)
        k = int(d1 * d2 * top_k_ratio)

        topk_values, topk_indices = torch.topk(magnitude.view(-1), k)
        self.mask = torch.zeros((d1, d2), dtype=torch.bool)
        self.mask.view(-1)[topk_indices] = True

        self.c = nn.Parameter(W_freq[self.mask].clone().detach(), requires_grad=True)

    def forward(self, x):
        F = torch.zeros(self.base_layer.weight.shape, dtype=torch.complex64, device=x.device)
        F[self.mask] = self.c
        Delta_W = torch.fft.ifft2(F).real * self.alpha

        h = self.base_layer(x)
        if x.dim() == 2:
            h += torch.matmul(x, Delta_W)
        elif x.dim() == 3:
            h += torch.einsum('bnd,df->bnf', x, Delta_W)
        return h
```

In this section, we introduce our proposed Layerwise Fourier Masked Adapter (LFMA), a parameter-efficient method for adapting models to downstream tasks. By leveraging Fourier transforms, we sparsify weight updates in the frequency domain, selecting and optimizing only the most significant components. This approach reduces computational overhead while preserving performance.

3.1 Fourier Layer Adaptation

The core of our method is the LFMA module, which fundamentally leverages the properties of the Fourier transform to facilitate efficient and targeted layer-wise adaptation in neural networks. At its core, LFMA encapsulates the idea that weight perturbations $\mathbf{W}$. $\mathbf{W} \in \mathbf{R}^{d_1 \times d_2}$ can be effectively represented and manipulated within the frequency domain (32).The adaptation process involves constructing a small perturbation matrix ΔW_{init}, which is transformed into the frequency domain:

$$\mathbf{W}_{\text{freq}} = \mathcal{F}(\Delta \mathbf{W}_{\text{init}}), \tag{1}$$

where $\mathcal{F}(\cdot)$ denotes the FFT (6), yielding complex-valued coefficients. To promote spectral sparsity and focus the adaptation on the most influential frequencies, a top-k selection based on the magnitude of these coefficients is performed (5; 33). The set of coefficients corresponding to the mask is then designated as learnable parameters c, which are optimized during training. The adapted output in the spatial domain is reconstructed via the inverse Fourier transform $\mathcal{F}^{-1}$, scaled by a hyperparameter $a > 0$, ensuring that the spectral modifications are smoothly integrated into the original weights. This process emphasizes spectral regularization, constraining the adaptation to a compact set of critical frequency components, which effectively mitigates overfitting and enhances interpretability (34). The LFMA module thus provides a principled mechanism for layer-specific, frequency-aware fine-tuning, aligning with the signal processing intuition that the most salient features are often concentrated in a subset of the spectral domain.

3.2 Inverse Fourier and Top-K Selection

In this section details the forward pass of the adapter, which reconstructs the weight perturbation via the inverse Fourier transform, integrates only the real part into the computation, and optimizes the selected top-k components during training. reconstructing a sparse frequency matrix, initialized to zeros.

top k ratio (Params)	SST-2 (Acc.)	MRPC (Acc.)	QNLI (Acc.)	RTE (Acc.)	CoLA (MCC)	STS-B (PCC)	Avg.
FF (125M)	94.8	90.2	92.8	78.7	63.6	91.2	**85.2**
LoRA (0.3M)	**95.1** ± 0.2	89.7 ± 0.7	**93.3** ± 0.3	78.4 ± 0.8	63.4 ± 1.2	**91.5** ± 0.2	**85.2**
FourierFT (0.024M)	94.2 ± 0.3	90.0 ± 0.8	92.2 ± 0.1	79.1 ± 0.5	**63.8** ± 1.6	90.8 ± 0.2	85.0
0.0016 (0.023M)	**94.4** ± 0.3	**90.5** ± 0.3	92.2 ± 0.2	**80.9** ± 0.2	60.1 ± 1.0	90.4 ± 0.3	84.8
0.001 (0.015M)	**94.4** ± 0.2	**90.5** ± 0.3	92.0 ± 0.1	79.4 ± 0.8	60.0 ± 1.0	90.2 ± 0.3	84.4
0.0005 (0.007M)	93.4 ± 0.2	88.8 ± 0.5	90.6 ± 0.9	78.1 ± 1.9	57.9 ± 1.1	88.2 ± 0.6	82.8

Table 2: Performance comparison on the GLUE benchmark with RoBERTa-Base. Accuracy is used for SST-2, MRPC, QNLI, RTE, and MCC and PCC are used for CoLA and STS-B respectively. We report the main performance score using median among five different experimentation varying in seeds setting. Avg. is average of six benchmark datasets.

top k ratio (Params)	SST-2 (Acc.)	MRPC (Acc.)	QNLI (Acc.)	RTE (Acc.)	CoLA (MCC)	STS-B (PCC)	Avg.
FF (356M)	96.4	90.9	94.7	86.6	68	**92.4**	**88.2**
LoRA (0.8M)	96.2 ± 0.5	90.2 ±1.0	**94.8** ± 0.3	85.2 ± 1.1	**68.2** ± 1.9	92.3 ± 0.5	87.8
FourierFT (0.048M)	96.0 ± 0.2	90.9 ± 0.3	94.4 ± 0.4	**87.4** ± 1.6	67.1 ± 1.4	91.9 ± 0.4	88.0
0.0005 (0.025M)	**96.5** ± 0.4	**91.0** ± 0.5	93.8 ± 0.2	85.1 ± 1.3	65.8 ± 0.3	90.9 ± 0.3	87.2
0.0003 (0.015M)	95.5 ± 0.3	89.5 ± 0.9	92.1 ± 0.2	83.0 ± 1.1	64.8 ± 1.1	90.3 ± 0.6	85.9
0.0001 (0.005M)	93.9 ± 0.3	88.2 ± 1.1	89.5 ± 0.3	78.3 ± 1.5	62.1 ± 1.2	86.6 ± 1.0	83.1

Table 3: Performance comparison on the GLUE benchmark with RoBERTa-Large. Accuracy is used for SST-2, MRPC, QNLI, RTE, and MCC and PCC are used for CoLA and STS-B respectively. We report the main performance score using median among five different experimentation varying in seeds setting. Avg. is average of six benchmark datasets.

The inverse Discrete Fourier Transform (IDFT) is then applied to recover the spatial-domain perturbation:

$$\Delta \mathbf{W} = R\left(\mathcal{F}^{-1}(\mathbf{F})\right) \cdot \alpha \tag{2}$$

where $R(\cdot)$ extracts the real number part, and $\mathcal{F}^{-1}(\cdot)$ is the inverse FFT. This ensures that only a sparse subset of frequencies contributes to the adaptation, significantly reducing the parameter. During fine-tuning, only the $\mathbf{c}$ parameters in each adapter are optimized while the rest of the model remains frozen. This approach exploits the low-rank structure of weight perturbations in the Fourier domain, inspired by prior work on spectral methods in neural networks (3; 31). Hyperparameters such as α control the adaptation strength and sparsity, respectively.The adapted output is computed by adding this perturbation to the base layer's computation. For an input tensor x, the base layer produces $h = \text{layer}(x)$. The perturbation is then integrated as follows:

- If $\mathbf{x}$ is 2D (e.g., batch size $\times$ input dimension), $\mathbf{h} \leftarrow \mathbf{h} + \mathbf{x} \cdot \Delta\mathbf{W}$.
- If $\mathbf{x}$ is 3D (e.g., batch size $\times$ sequence length $\times$ input dimension), $\mathbf{h} \leftarrow \mathbf{h} + \text{einsum}(\mathbf{x}, \Delta\mathbf{W})$, using efficient tensor contraction.

During training, only the top-k selected components are optimized via backpropagation, as they are the learnable parameters. This top-k learning strategy ensures that gradients flow exclusively through the most impactful frequencies, further enhancing efficiency. The mask remains fixed after initialization, preventing unnecessary exploration of low-magnitude components.

The pseudocode for LFMA is shown as Algorithm 1 in PyTorch style.

4 Experiments

In this section, we first evaluate the overall performance of our proposed method, Layerwise Fourier Masked Adapter (LFMA), by comparing it against several strong baseline methods across domains of Natural Language Processing (NLP) and Computer Vision (CV). We then conduct an ablation study to analyze the impact of our key hyperparameter, the `top_k_ratio`, which controls the parameter efficiency of our approach.

4.1 Image Classification

Datasets. To rigorously evaluate the efficacy and generalization capabilities of our proposed method, Layerwise Fourier Masked Adapter (LFMA), we conduct experiments on a comprehensive suite of seven widely-used image classification benchmarks. This collection includes generic object

recognition datasets (**CIFAR-10**, **CIFAR-100** (35)), fine-grained visual classification tasks (**Oxford-IIIT Pet** (36), **Stanford Cars** (37), **FGVC-Aircraft** (38)), and remote sensing imagery datasets (**EuroSAT** (39), **RESISC45** (40)). The diversity of these benchmarks allows us to thoroughly assess the robustness and adaptability of LFMA across various data domains and scales.

Compared Schemes. We benchmark LFMA against several established fine-tuning paradigms. The performance of **Full Fine-Tuning (Full-FT)** is presented as a practical upper bound. We compare against **Low-Rank Adaptation (LoRA)** (2), a prominent and widely-adopted parameter-efficient fine-tuning (PEFT) technique. Furthermore, we include **FourierFT** (4), a state-of-the-art method operating in the frequency domain, which serves as our most direct and challenging baseline.

Implementation Details. Our experiments leverage Vision Transformer (ViT) backbones (13), specifically **ViT-Base (ViT-B/16)** and **ViT-Large (ViT-L/16)**, both pre-trained on ImageNet-21k (41). For model training, we employ the AdamW optimizer (42) with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-4}. A consistent batch size of 32 is used, and all input images are resized to a resolution of 224x224. No learning rate scheduler is applied to ensure a fair comparison of the models' intrinsic learning capabilities. For our LFMA method, the adapter is integrated into the query projection matrix within the self-attention mechanism of every transformer block. Based on empirical validation, the primary hyperparameters for LFMA are set to a Fourier coefficient scaling factor of $\alpha = 120$ and a top-k masking ratio of 0.0003. Deviations from this configuration for specific experiments, such as those presented in Table 1, are explicitly noted.

Main Results. The comparative performance of our LFMA model variants is summarized in **Table 1**. The results, obtained with specific hyperparameters ($\alpha = 12$, top-k=0.05), demonstrate the strong performance of our approach. The LFMA (ViT-B) variant achieves the highest accuracy on five of the seven datasets, establishing its efficacy as a robust general-purpose adapter. Notably, the LFMA (ViT-L) variant shows competitive or superior performance on challenging fine-grained datasets like Stanford Cars and FGVC-Aircraft, suggesting its particular aptitude for tasks requiring detailed feature discrimination. These results collectively validate that LFMA provides a compelling and parameter-efficient alternative to full fine-tuning.

4.2 Natural Language Understanding

Datasets. To assess the effectiveness of our proposed LFMA in the natural language domain, we evaluate on the GLUE benchmark (43), a comprehensive suite designed to measure various aspects of Natural Language Understanding (NLU). Specifically, we include SST-2, MRPC, QNLI, RTE, CoLA, and STS-B, while excluding QQP, WNLI, and MNLI to compare with the baseline. We adopt the standard GLUE benchmark splits for training, validation, and testing, ensuring consistency with widely used experimental setups.

Compared Schemes. We benchmark LFMA against three strong baselines including Full Fine-tuning (FF) and parameter-efficient fine-tuning, LoRA (2) and FourierFT (4). We directly adopt the reported results of baselines from the FourierFT paper. Together, these baselines provide a comprehensive evaluation framework, as LoRA is the most commonly used PEFT method and FourierFT is the conceptually related approach to LFMA.

Implementation Details. We experiment with two models, which have been used in FourierFT paper, RoBERTa-Base and RoBERTa-Large (14). We utilized the AdamW optimizer (42), with hyperparameters (learning rate, epochs, batch size, seeds, and scaling factor α) based on those reported in the FourierFT (4) paper, making slight adjustments during experimentation to better fit our setting. We followed the experimental setup applied in LoRA (2) to conduct fair comparison with FourierFT, which is the main baseline, LFMA adapters are applied to the query and value projections in each transformer block as well. The top_k_ratio controls the proportion of frequency components retained for fine-tuning, and we evaluate multiple ratios as reported in Tables 2 and 3.

Main Results. Table 2 and Table 3 present the GLUE benchmark results for RoBERTa-Base and RoBERTa-Large, respectively. Accuracy is used for SST-2, MRPC, QNLI, RTE, and MCC and PCC are used for CoLA and STS-B respectively. We report the main performance score using median among five different experimentation varying in seeds setting. Avg. is average of six benchmark datasets. Across both backbones, LFMA demonstrates that parameter-efficient fine-tuning in the Fourier domain is highly effective for natural language understanding tasks. Overall, LFMA achieves

performance comparable to or better than the FourierFT (4) baseline while using significantly fewer trainable parameters, thereby confirming its scalability and robustness beyond vision tasks.

For **RoBERTa-Base** (Table 2), LFMA outperforms both LoRA and FourierFT (4) on MRPC and RTE, while maintaining performance comparable to or exceeding FourierFT on all datasets except CoLA. This highlights LFMA's effectiveness in sentence-pair classification tasks. Although performance slightly decreases at the lowest top-k ratios, the method consistently delivers strong results with far fewer parameters, validating its efficiency. However, we note that variance across runs increases as the parameter budget becomes smaller.

For **RoBERTa-Large** (Table 3), LFMA surpasses both baselines on SST-2 and MRPC, and delivers performance comparable to or better than FourierFT (4) on most other datasets, again with the exception of CoLA. These results indicate that LFMA scales effectively to larger backbones, retaining competitiveness even under strict parameter budgets. An additional observation is that as the number of trainable parameters decreases, the variance in performance tends to increase more noticeably than in the Base model, underscoring a trade-off between parameter efficiency and stability at scale.

4.3 Ablation Study

To analyze the impact of the `top_k_ratio` hyperparameter, which directly controls the number of fine-tuned frequency components, we conducted an ablation study. The number of trainable parameters is directly proportional to the `top_k_ratio`. In the image classification task, for the ViT-Base model, a `top_k_ratio` of 0.1, 0.05, and 0.0003 corresponds to 707,784, 353,892, and 2,112 trainable adapter parameters, respectively. For the ViT-Large model, these ratios result in 2,516,568, 1,258,272, and 7,536 trainable parameters. Our main results were achieved with a `top_k_ratio` of 0.0003, which isolates only a tiny fraction of the most significant frequency components. In the NLU task, we further examined the trade-off between parameter count and performance. Notably, for the RoBERTa-Base model, LFMA achieves competitive results with FourierFT (4) while using a nearly identical number of parameters. In the RoBERTa-Large setting, LFMA achieves comparable or even superior results to FourierFT (4) despite requiring only about half the number of parameters. Moreover, as the number of trainable parameters decreases, the decrease in performance remains modest rather than drastic. The strong performance obtained with this setting underscores our core hypothesis: a substantial portion of the frequency spectrum is redundant for task-specific adaptation, and focusing on a small, highly informative subset is sufficient for effective fine-tuning. This selective update strategy is the key to LFMA's exceptional parameter efficiency, allowing it to push the boundaries of PEFT by achieving high performance with minimal computational overhead.

5 Conclusion

In this work, we presented the LFMA, an innovative adapter module that enhances efficient fine-tuning of pre-trained models through frequency-domain sparsity. By applying Fourier transforms to initial weight perturbations, selecting top-k high-magnitude frequency components for optimization, and integrating only the real part of the inverse transform into the forward pass, our method achieves a sparse yet effective adaptation strategy. This approach not only minimizes trainable parameters but also leverages signal processing principles to separate global and local adaptations, offering improved interpretability over traditional methods. Our experiments demonstrate LFMA is better than FourierFT (4). Across benchmarks in natural language understanding (e.g., GLUE (43)) and image classification (e.g., CIFAR-10/100 (35)), our adapter consistently outperforms FourierFT (4) with similar or lower parameters.

5.1 Future work

For the future work, we aim to investigate the way to use dynamic masking, allowing the mask to adapt during training based on real-time updating (**?**). Second, extend the method to various models like Diffusion model (44), Graph Neural Network (GNN) (45) across domains. Third, we will research for theory of the convergence properties that are appropriate for top-k frequency optimization, solving bounds, and getting error lower. Finally, deploying the adapter in real-world settings, such as continual learning (46).

References

[1] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., & Gelly, S. (2019). Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*.

[2] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. (2022). LoRA: Low-rank adaptation of large language models. *ICLR*.

[3] Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2020). Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*.

[4] Gao, Z., Wang, Q., Chen, A., Liu, Z., Wu, B., Chen, L., & Li, J. (2024). Parameter-efficient fine-tuning with discrete fourier transform. *arXiv preprint arXiv:2405.03003*.

[5] Park, Y., Jang, J., & Kang, U. (2020). Fast partial fourier transform. *arXiv preprint arXiv:2008.12559*.

[6] Wang, Z. (1984). Fast algorithms for the discrete W transform and for the discrete Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(4), 803–816.

[7] Kwon, K. S., & Lin, R. M. (2004). Frequency selection method for FRF-based model updating. *Journal of Sound and Vibration*, 278(1-2), 285–306.

[8] Stuchi, J. A., Boccato, L., & Attux, R. (2020). Frequency learning for image classification. *arXiv preprint arXiv:2006.15476*.

[9] Zhang, T., Ye, W., Yang, B., Zhang, L., Ren, X., Liu, D., Sun, J., Zhang, S., et al. (2022). Frequency-aware contrastive learning for neural machine translation. In *Proceedings of the AAAI conference on artificial intelligence*, 36(10), 11712–11720.

[10] Zhou, M., Huang, J., Yan, K., Hong, D., Jia, X., Chanussot, J., & Li, C. (2024). A general spatial-frequency learning framework for multimodal image fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[11] Tang, C., Wang, X., Bai, Y., Wu, Z., Zhang, J., & Huang, Y. (2023). Learning spatial-frequency transformer for visual object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(9), 5102–5116.

[12] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877–1901.

[13] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Gelly, S. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[14] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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

[16] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune BERT for text classification?. *arXiv preprint arXiv:1905.05583*.

[17] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

[18] Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., ... & Dean, J. (2022). The carbon footprint of machine learning. *ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies (COMPASS)*.

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

[20] Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 4582–4597.

[21] Mallick, S., Bhowmik, S., & Pal, M. (2023). LoRA-ViT: A parameter-efficient approach to fine-tuning vision transformers. *arXiv preprint arXiv:2308.13931*.

[22] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv preprint arXiv:2305.14314*.

[23] Zhang, Q., Chen, M., Bukharin, A., He, P., Lee, Y., Chen, L., ... & Zhao, R. Y. (2023). AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning. *ICLR*.

[24] Liu, S. Y., Lin, C. Y., Lee, C. Y., & Lee, H. Y. (2024). DoRA: Weight-Decomposed Low-Rank Adaptation. *arXiv preprint arXiv:2402.09353*.

[25] He, J., Radev, D., & Neubig, G. (2022). Towards a unified view of parameter-efficient transfer learning. *ICLR*.

[26] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Goyal, N. (2023). LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

[27] Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

[28] Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., & Tang, J. (2021). P-tuning: Prompt-based tuning is comparable to fine-tuning across scales and tasks. *arXiv preprint arXiv:2103.10385*.

[29] Le, Q., Sarlós, T., & Smola, A. (2013). Fastfood: Approximate kernels via fourier and hadamard transforms. *International Conference on Machine Learning (ICML)*.

[30] Zhang, S., & Zhang, X. (2018). F-conv: A fast and efficient convolution operation for deep learning. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, 369–377.

[31] Wang, H., Zhang, Z., Liu, N., & Wang, M. (2022). Spectral-based graph neural networks. *arXiv preprint arXiv:2209.11155*.

[32] Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., ... & Courville, A. (2019). On the spectral bias of neural networks. In *International Conference on Machine Learning*, 5301–5310.

[33] Ozay, M., & Okatani, T. (2019). Learning sparse features in deep neural networks using a fourier-based approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10), 2455–2468.

[34] Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations (ICLR)*.

[35] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. *University of Toronto*.

[36] Parkhi, O. M., Vedaldi, A., Zisserman, A., & Jawahar, C. V. (2012). Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, 3498–3505.

[37] Krause, J., Stark, M., Deng, J., & Fei-Fei, L. (2013). The stanford cars dataset. In *3D-RR*.

[38] Maji, S., Raichur, E., Urtasun, R., & Darrell, T. (2013). Fine-grained visual classification of aircraft. *British Machine Vision Conference (BMVC)*.

[39] Helber, P., Bischke, B., Dengel, A., & Borth, D. (2019). EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7), 2217–2226.

[40] Cheng, G., Han, J., & Lu, X. (2017). Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10), 1865–1883.

[41] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255.

[42] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

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

[44] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33, 6840–6851.

[45] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., ... & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1, 57–81.

[46] Shin, H., Lee, J. W., Kim, J., & Kim, J. (2017). Continual learning with deep generative replay. *Advances in Neural Information Processing Systems*, 30.