S^{2}FT: Parameter-Efficient Fine-Tuning in Sparse Spectrum Domain

Baoquan Zhang[1], Zhehao Yu[1], Lisai Zhang[3], Kenghong Lin[*1], Tianran Chen[1], Yuxi Sun[4], Yunming Ye[*1], Yao He[2]
[1] Harbin Institute of Technology, Shenzhen; [2] ShenZhen SiFar Co., Ltd.; [3] Bilibili. Inc; [4] Shenzhen University
baoquanzhang@hit.edu.cn, {210110629, linkenghong}@stu.hit.edu.cn, lisaizhang@foxmail.com,
23S151121@stu.hit.edu.cn, sunyuxi@szu.edu.cn, yeyunming@hit.edu.cn, heyao18818@gmail.com

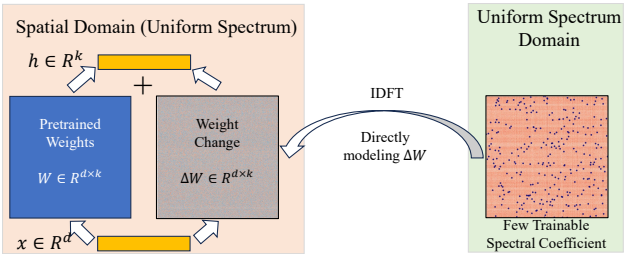
Abstract

Parameter Efficient Fine-Tuning (PEFT) is a key technique for adapting a large pretrained model to downstream tasks by fine-tuning only a small number of parameters. Recent methods based on Fourier transforms have further reduced the fine-tuned parameters scale by only fine-tuning a few spectral coefficients. Its basic assumption is that the weight change ΔW is a spatial-domain matrix with a sparse spectrum. However, in this paper, we observe that the spectrum of weight change is not sparse, but instead distributed like power-uniform. This fact implies that fine-tuning only a few spectral coefficients is insufficient to accurately model the weight change with uniform spectrum. To address this issue, we propose to seek an invertible transformation that can transform a latent spatial-domain matrix with sparse spectrum to the weight change, and then perform PEFT on such sparse spectrum domain with few spectral coefficients, called S^{2}FT. To seek such transformation, we first pre-estimate a coarse weight change as a prior. Then, inspired by that sparse spectrum often correspond to locally smooth spatial structures, we regard this transformation as a row and column rearrangement operation on the pre-estimated weight change that smooth spatial structures while keep the structure information of neurons. Finally, we propose to solve the rearrangement search problem in a simple nearest neighbor search manner, thereby obtaining the invertible transformation. Extensive results show our S^{2}FT achieves superior performance by only using 0.08% training parameters.

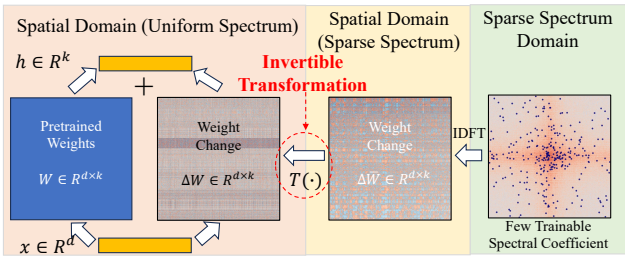
1. Introduction

Pretrained large models have gained significant attention in computer vision and natural language processing [7, 17, 24, 28, 49, 50]. In earlier works, full fine-tuning is commonly used to adapt these pretrained models to specific downstream tasks. However, as the model size increases, the computational resources required for full fine-tuning and parameter storage become a limiting factor. To address this challenge, Parameter Efficient Fine-Tuning (PEFT) was introduced recently and has received widespread interest, which aims to adapt a pretrained large model to downstream tasks by fine-tuning only few trainable parameters [15, 19, 29, 53, 56].

Low-Rank Adaptation (LoRA) [19] is a popular method for PEFT, which aims to model the weight change ΔW by using a product between two low-rank matrices A and B as $\Delta W = AB$. The parameter scale of the low-rank matrix is much smaller than that of the original parameters. Despite LoRA's superior performance, the parameter scale for large models is still heavy, imposing storage burden for both public communities and individual users. Recently,



(a) Existing FourierFT.



(b) Our S^{2}FT.

Figure 1. Existing FourierFT (a) directly models weight change ΔW by few trainable spectral coefficients, the performance is limited by the non-frequency-sparse nature of ΔW. Our S^{2}FT (b) seeks an invertible transformation from a latent spatial-domain matrix $\Delta \bar{W}$ with sparse spectrum to the ΔW, and perform PEFT on the sparse spectrum domain with fewer spectral coefficients. (In the rightmost plot, the orange depth indicates the power spectrum, and blue dots denote trainable spectral coefficients.)

*Corresponding author.

Gao et al. [9] further decrease the parameter scale by a novel Fourier transform-based method called FourierFT. As shown in Figure 1a, it directly regards ΔW as a spatial-domain matrix, and then achieves PEFT by fine-tuning on spectral domain with a few trainable spectral coefficients. The basic assumption of such design is that the weight change ΔW is a spatial domain matrix with a sparse spectrum. However, a natural question is that *is such assumption really reasonable?*

To answer this question, we conduct a detailed analysis of the weight change ΔW, and observe that when it is directly treated as a spatial-domain matrix, its spectrum is typically not sparse but instead exhibits a power-uniform distribution (see Section 3.1.2 for details). Under such a distribution, it is difficult to identify a small number of spectral coefficients that can accurately reconstruct the original weight change, which limits PEFT's effectiveness. Ideally, Fourier-based PEFT perform best when the underlying ΔW is smooth in spatial domain, corresponding to a sparse spectrum that can be well-approximated using only a few spectral coefficients.

Motivated by this, we propose a novel PEFT approach in the sparse spectral domain, termed S^2FT. As shown in Figure 1b, instead of perform PEFT on the ΔW's uniform spectrum, we propose to seek an invertible transformation that maps a smooth latent matrix $\Delta\bar{W}$ —whose spectrum is sparse—into ΔW and then perform PEFT on the spectrum-sparse latent matrix $\Delta\bar{W}$ by optimizing only a few of its spectral coefficients. The key advantage of this approach lies in its ability to fully leverage the expressive power of a sparse spectrum and accurately model ΔW with fewer trainable parameters. The key challenge is to discover an appropriate invertible transformation. To tackle this, we first pre-estimate a coarse weight change $\Delta\hat{W}$ via gradient accumulation on a small training subset, serving as a prior. Then, inspired by the well-known principle that sparse spectra often correspond to locally smooth spatial structures, we formulate the invertible transformation as a solution of row and column rearrangement problem on $\Delta\hat{W}$ that minimize the distance between adjacent rows and columns. The rearrangement should smooth the weight matrix while do not disrupt the structure information of the neurons. Finally, we propose to solve the rearrangement problem in a simple nearest neighbor search manner, thereby obtaining the invertible transformation for achieving our S^2FT.

Our main contributions can be summarized as follows:

- We experimentally confirm that the spectrum of weight change is not sparse but rather tends to be power-uniform. This indicates that fine-tuning only a few spectral coefficients is insufficient for accurately modeling ΔW, which limits the performance of PEFT. To the best of our knowledge, this is the first work to identify this challenge.
- We propose a novel PEFT method in sparse spectrum domain, which seeks an invertible transformation that can transform a latent spatial-domain matrix with sparse spectrum to the weight change, and then perform PEFT on such sparse spectrum domain. Its advantage is that a few spectral coefficients can be fully exploited for PEFT.
- We conduct experiments on various tasks and datasets, which verify the effectiveness of our S^2FT.

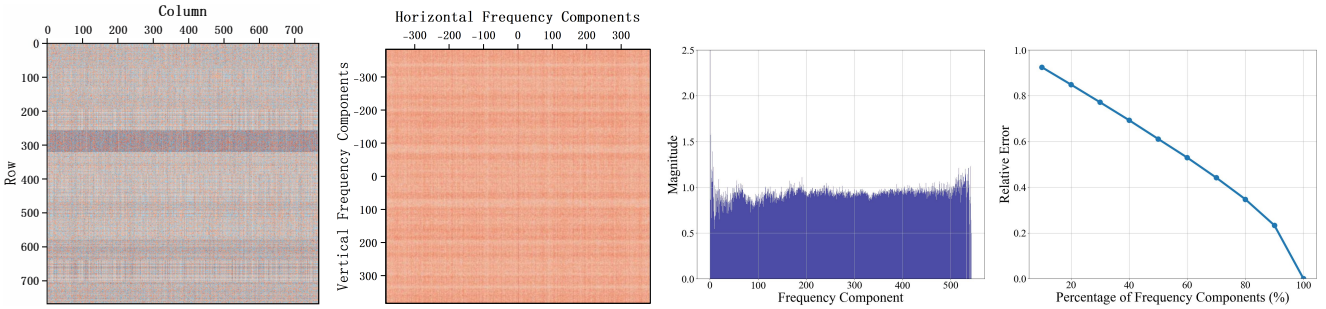
2. Related Works

2.1. Parameter-Efficient Fine-Tuning (PEFT)

PEFT [30] is a challenging task, which aims to tune only few trainable parameters to achieve adaptation of pretrained models on downstream tasks. Existing methods can be roughly categorized into three groups: **(1) Reparameterization-based Methods.** These methods primarily center on directly modeling the weight changes using few trainable parameters and executing PEFT through reparameterization techniques[4, 12, 19, 20, 25–27, 44, 54]. For instance, Gao et al. [9] employed a Fourier-domain transformation, referred to as FourierFT, under the assumption that weight changes can be represented as a spatial-domain matrix with a sparse spectrum. This enables PEFT by fine-tuning only a select few spectral coefficients. **(2) Addition-based Methods.** This class of methods achieves PEFT by integrating additional adapters or optimizing prompts and prefixes within pre-trained layers [3, 13, 18, 33, 38, 46]. **(3) Selection-based Methods**. This category emphasizes tuning only a subset of the pre-trained parameters to achieve PEFT [15, 16, 18, 43, 47, 53]. In this paper, we focus on Fourier transformation-based methods. Differently, instead of directly fine-tuning coefficients on a uniform spectrum domain, we propose to perform fine-tuning on a sparse spectrum domain so that fewer trainable coefficients are required for further enhancing PEFT performance.

2.2. Fourier Transform for Deep Learning

Fourier transform plays a fundamental role across information processing, providing an effective way to extract frequency components. Fourier transform has been used for representation learning [8, 11, 36, 45]. For example, in [6, 31], Fourier transform is used to convert spatial-domain information into frequency representations to extract discriminative patterns from challenging datasets containing noise or high dimensionality. Recently, Gao et al. [9] propose viewing the weight change matrix ΔW as a spatial-domain matrix and performing PEFT by applying a Fourier transform, thereby fine-tuning only a small number of its spectral coefficients. However, in this paper, we observe that the spectrum of the weight change is not inherently sparse, but instead exhibits a power-uniform distribution. To address this, we introduce a latent spatial-domain matrix that yields a sparse spectrum through an invertible transformation. By fine-tuning the spectral coefficients derived from this sparsified representation, our method enables more effective PEFT.



(a) Weight change ΔW. (b) Power spectrum of ΔW. (c) Amplitude distribution. (d) Relative error of ΔW.

Figure 2. Distribution analysis of weight change ΔW in spatial and spectral domains.

3. Methodology

3.1. Preliminaries and Motivation Analysis

3.1.1. Preliminaries: FourierFT

Formally, let W denotes a pre-trained weight matrix, i.e., $W \in \mathbb{R}^{d\times k}$ and ΔW is its weight change after adapting to downstream tasks. The key challenge of achieving PEFT is how to model the ΔW by using only few trainable parameters. Recently, Gao et al. [9] effectively address this challenge by a Fourier transform-based method (FourierFT). Specifically, they firstly regard the weight change ΔW as a spatial-domain matrix. Then, they randomly initialize a spectral-domain matrix $F \in \mathbb{C}^{d\times k}$ where $\mathbb{C}$ denotes complex field and only few spectral coefficients are nonzero (i.e., trainable parameters) and other are zero. After that, the weight change ΔW can be represented as a spatial-domain matrix obtained by using an inverse Fourier transform on spectral-domain matrix $F \in \mathbb{C}^{d\times k}$:

$$\Delta W = \mathcal{R}(IDFT(F)) * \alpha, \tag{1}$$

where $\mathcal{R}(\cdot)$ denotes an operation taking the real part of the complex matrix, $IDFT(\cdot)$ is the inverse Fourier transform, and α is a hyper-parameter. Finally, the adapted weight W' can be represented as:

$$W' = W + \Delta W = W + \mathcal{R}(IDFT(F)) * \alpha. \tag{2}$$

The assumption of leveraging few spectral coefficients to model the weight change ΔW is that its spectrum is very sparse. However, a natural question is that *is such assumption really reasonable?*

3.1.2. Motivation Analysis on Weight Change $\triangle W$

To answer the above question, we conduct a detailed analysis on weight change $\triangle W$. Specifically, we randomly select a weight matrix W from a pretrained vision transformer (ViT) model and leverage full fine-tuning to calculate its weight change $\triangle W$. Then, we visualize $\triangle W$ and its power spectrum in Figures 2a and 2b (please see Appendix for more results). It can be clearly observed that (1) the distribution of $\triangle W$ is chaotic in spatial domain (see Figure 2a), which means that it contains various frequency components; (2) the power distribution is close to uniform in the entire spectral domain (see Figure 2b). Furthermore, we visualize the amplitude at different frequency components in Figure 2c. It can be found that the power spectrum of weight change ΔW is indeed not sparse, but tends to be power-uniform. This suggests that modeling $\triangle W$ using few spectral coefficients is very difficult, which limits the PEFT performance of FourierFT. To further verify this observation, we visualize the average relative reconstruction error of every $\triangle W$ in ViT in Figure 2d where we gradually sample its spectral coefficients from 10% to 100%. We notice that it requires to sample over 90% spectral coefficients to achieve less than 10% reconstruction error.

The above analysis indicates a fact that directly using few spectral coefficients is insufficient to model the weight change ΔW since it is a spectrum-uniform matrix. Thus, an intuitive insight is *whether there is a potential spatial-domain matrix $\Delta\bar{W}$ with sparse spectrum which can be non-destructively transformed to the weight change ΔW?* If it exists, we can perform PEFT on this sparse spectrum such that few spectral coefficients can be fully exploited for modeling the weight change ΔW.

3.2. PEFT in Sparse Spectrum Domain (S^{2}FT)

Based on the above insight, we propose a novel parameter-efficient fine-tuning method in sparse spectrum domain as shown in Figure 1b, called S^{2}FT. Instead of directly modeling the weight change ΔW with an uniform spectrum, we attempt to seek an invertible transformation that can transform a latent spatial-domain matrix $\Delta\bar{W}$ with sparse spectrum to the weight change ΔW, and then perform PEFT in such sparse spectrum space with few spectral coefficients. To formulate, given a pretrained weight W and a downstream task $\tau = \{\mathcal{D}_{tr}, \mathcal{D}_{te}\}$ where $\mathcal{D}_{tr}$ and $\mathcal{D}_{te}$ denote its training and test set, we first pre-estimate a coarse weight change $\Delta\hat{W}$ for each weight W as a prior. Then, we leverage the prior $\Delta\hat{W}$ to seek an invertible transformation that can transform a latent spatial-domain matrix $\Delta\bar{W}$ with sparse spectrum to the weight change ΔW. Finally, we perform PEFT in such sparse spectrum domain, where the adapted weight W' on

downstream tasks can be expressed as:

$$\begin{aligned} W' &= W + \Delta W \\ &= W + T(\Delta \bar{W}) \\ &= W + T(\mathcal{R}(IDFT(F)) * \alpha). \end{aligned} \tag{3}$$

where $T(\cdot)$ denotes the invertible transformation from a latent spatial-domain matrix $\Delta\bar{W}$ with sparse spectrum to the weight change ΔW; F is a sparse complex matrices, i.e., only few values are learnable coefficients and others are zeros (see Section 3.2.3 for its allocation details). Next, we introduce the details of the weight change pre-estimation, invertible transformation, and spectral coefficient sampling strategy on Sections 3.2.1, 3.2.2, and 3.2.3, respectively.

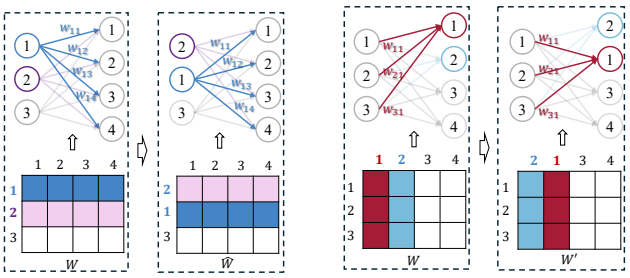
3.2.1. Weight Change Pre-estimation

In order to obtain the proper transformation $T(\cdot)$, we need to pre-estimate the pattern of ΔW first and then leverage it to guide us to seek the invertible transformation $T(\cdot)$. An ideal estimation way is first leveraging full fine-tuning to obtain the adapted weight W' with the entire training set $\mathcal{D}_{tr}$ and calculate the weight change $\Delta\hat{W}$ by using their difference, i.e., $\Delta\hat{W} = W' - W$. However, such method is 1) infeasible since it consumes huge computing cost; and 2) is contradictory to PEFT since the goal of PEFT is to reduce the full fine-tuning's training cost. Interestingly, we find that an very accurate estimation of ΔW is not required; actually, even a coarse approximation can also effectively guide the search for this invertible transformation $T(\cdot)$ (see Table 2 in Appendix). To this end, we propose a simple yet efficient pre-estimation method that leveraging the negative cumulative gradient of a subset $\mathcal{D}_{sub}$ from training set $\mathcal{D}_{tr}$ to pre-estimate a coarse weight change $\Delta\hat{W}$:

$$\Delta\hat{W} = -\sum_{x \in \mathcal{D}_{sub}} \nabla\mathcal{L}_x(w_i), \tag{4}$$

where $\nabla\mathcal{L}_x(W)$ is the gradient of weight W from training sample x. We note that some PEFT works also calculate gradient over all training data [15, 53], however, different from these methods that leverage gradient to estimate weight importance, we leverage it to pre-estimate weight change.

Why can the negative cumulative gradient roughly pre-estimate the weight change $\Delta\hat{W}$? In fact, the weight change $\Delta\hat{W}$ can be viewed as the direction of model weight update, i.e. moving the pretrained weights along such direction can minimize the loss on the training samples of a downstream task. According to the theory of gradient descent, we know that updating model parameters along the opposite direction of the cumulative parameter gradient can effectively reduce the model loss on training samples. This means that the opposite direction of cumulative gradient on training samples is exactly the weight update direction we are looking for. Thus, the negative cumulative gradient can be used to roughly pre-estimate the weight change $\Delta\hat{W}$.



(a) Switch rows 1 and 2 of W. (b) Switch columns 1 and 2 of W.

Figure 3. Rows/columns rearrangement only affects the order of inputs and outputs, without disrupting the link structure information between input and output neurons. E.g., in (a), after switching the first two rows of W, the input neuron ① moves down and still associated with output neuron ② via w_{12}.

3.2.2. Invertible Transformation

The goal of this step is regarding the pre-estimate the weight change $\Delta\hat{W}$ as a guidance to seek an invertible transformation $T(\cdot)$ that can transform a latent spatial-domain matrix $\Delta\bar{W}$ with sparse spectrum to weight change ΔW. In general, the spectrum-sparse matrices typically exhibit local smoothness in the spatial domain. Thus, the transformation we are looking for needs to satisfy: **Point 1**) the weight change should be smoother after using such transformation; **Point 2**) the structure information of neuron (i.e., link dependence) should be preserved after applying such transformation; and **Point 3**) the transformation must be invertible since we need to accurately turn to the original weight space to perform PEFT. Fortunately, we find that the above three points can be well satisfied by the transformation of rows and columns rearrangement: 1) we can rearrange the rows and columns of weight change to minimizes the difference between adjacent rows or columns, thereby achieving **Point 1**; 2) as shown in Figure 3, rearranging the rows and columns of weight matrix does not destroy the link dependence of neurons, which satisfies **Point 2**; and 3) rows and columns rearrangement is a typical invertible transformation, i.e., satisfying **Point 3**.

Based on this, we formulate the invertible transformation as a solution of row and column rearrangement problem on the pre-estimated weight change $\Delta\hat{W} \in \mathbb{R}^{d\times k}$. The goal is to identify a permutation of rows (or columns) that minimizes the total distance between adjacent rows (or columns), thereby achieving a latent smoother spatial-domain matrix. While alternative approaches may exist, we leave the exploration of such methods to future work. Formally, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a complete weighted graph, where each node $v_i \in \mathcal{V}$ corresponds to a row (or column) of the matrix $\Delta\hat{W}$, and each edge $e_{ij} \in \mathcal{E}$ connects node v_i and v_j with an associated weight β_{ij}. The edge weight β_{ij} is defined as Euclidean distance between row i and row j. We aim to find a permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ of row (or column) indices that minimizes the total weight of adjacent pairs in

Algorithm 1 Nearest Neighbor Search (row rearrangement)

Input: Matrix $\Delta\hat{W} \in \mathbb{R}^{d\times k}$; Euclidean distance β_{ij}
Output: Permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$
Initialize set of unvisited indices $\mathcal{U} \leftarrow \{1, 2, \ldots, n\}$
Randomly select a starting index $i \in \mathcal{U}$ and set $\pi_1 \leftarrow i$
Remove i from $\mathcal{U}$
for $k \leftarrow 2$: n **do**
    Let $j^* = \arg\min_{j\in\mathcal{U}} \beta_{\pi_{k-1},j}$
    Set $\pi_k \leftarrow j^*$
    Remove j^* from $\mathcal{U}$
end for
return π

the rearranged sequence. That is,

$$\arg\min_{\pi\in\mathcal{P}_n} \sum_{k=1}^{n-1} \beta_{\pi_k,\pi_{k+1}}, \tag{5}$$

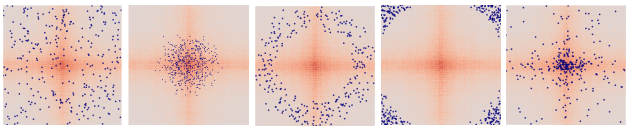
where $\mathcal{P}_n$ is the set of all permutations of $\{1, \ldots, n\}$ ($n = d$ when rearranging row otherwise $n = k$).

Proposition 1. For the rearranged matrix $\Delta\bar{W}$, the objective in Eq. (5) can be equivalently expressed as:

$$\begin{aligned}\sum_{k=1}^{n-1} \beta_{\pi_k,\pi_{k+1}} &= \sum_{k=1}^{n-1} |\Delta\bar{W}_{k+1} - \Delta\bar{W}_k|^2 \\ &= \sum_{u=0}^{n/2-1} 8\sin^2(\frac{\pi u}{n})|DFT(\Delta\bar{W})_u|^2,\end{aligned} \tag{6}$$

where $\Delta\bar{W}_k$ denotes the k-th row (or column) of $\Delta\bar{W}$, $DFT(\Delta\bar{W})_u$ denotes the u-th row (or column) of the spectral coefficients of $\Delta\bar{W}$. From the proposition, we can see that the objective in Eq. (5) corresponds to a weighted sum of spectral energies, where each frequency component is weighted by $8\sin^2(\frac{\pi u}{n})$. The weighting term $8\sin^2(\frac{\pi u}{n})$ grows monotonically with the frequency index u for $u \in [0, n/2-1]$ (for real-valued $\Delta\bar{W}$, the spectrum exhibits conjugate symmetry, so only the first half is considered), meaning that higher frequencies are assigned larger weights. Therefore, minimizing the left-hand side of Eq. (5) suppresses the energy of high-frequency components in $\Delta\bar{W}$ and forces most of its spectral energy to concentrate in the low-frequency. As a result, Eq. (5) implicitly encourages the rearranged matrix $\Delta\bar{W}$ to exhibit a sparse spectrum.

The Eq.5 is a typical NP-hard problem. In this paper, we adopt a nearest neighbor search algorithm [21] to solve it. As shown in Algorithm1, we iteratively select the unvisited node that is most similar to the current node (i.e., the one with the smallest distance β_{ij}), thereby constructing a permutation in a greedy manner. As a result, we can obtain a permutation π^r (or π^c) of the row (or column) indices, which is used to transform the pre-estimated weight change $\Delta\hat{W}$ into a spatial-domain matrix $\Delta\bar{W}$ with a sparse spectrum. It is important to note that this transformation merely involves re-indexing the rows and columns. Therefore, the invertible transformation $T(\cdot)$ in Eq. (3), which maps the spatial-domain matrix $\Delta\bar{W}$ back to the weight change ΔW, can be efficiently implemented via the inverse re-indexing operation, i.e., $\Delta W[\pi_r, :][:, \pi_c] = \Delta\bar{W}$.



(a) Random (b) LF (c) MF (d) HF (e) Ours

Figure 4. Distribution of spectrum and sampling points. LF/MF/HF bias sampling toward low/middle/high frequency.

3.2.3. Spectral Coefficient Sampling

The spectrum of S^{2}FT is sparse, which is different from previous FourierFT [9]. Therefore, a remaining question is should we use the same coefficient sampling strategy? After conduct existing sampling strategy of FourierFT (i.e., randomly sampling or sampling with a bias towards a favored frequency), we find that an interesting result different from FourierFT [9]: *S^{2}FT performs best when the spectral coefficients are sampled with a bias towards a low frequency, and such strategy is applicable to all evaluated tasks* (see Figure 1 in Appendix). To figure out the reason, we visualize the power spectrum of the latent spatial-domain matrix $\Delta\bar{W}$ and the sampling spectrum points in Figure 4. It can be seen that 1) our S^{2}FT indeed finds an invertible transformation that can transform the pre-estimated weight change $\Delta\hat{W}$ to a spatial-domain matrix $\Delta\bar{W}$ with sparse spectrum; and 2) among existing sampling strategies, biasing toward low frequencies (LF) performs best, but it still fails to accurately match the spectrum distribution (see Figure 4b).

Inspired by this observation, as shown in Figure 4e, we propose to regard the power-spectrum of $\Delta\bar{W}$ as a prior to estimate the sampling probability of spectral coefficient:

$$p(u, v) = \frac{\|DFT(\Delta\bar{W})\|^{\gamma}_{(u,v)}}{\sum_{u,v} \|DFT(\Delta\bar{W})\|^{\gamma}_{(u,v)}}. \tag{7}$$

where (u, v) denotes position, $0 \le u \le d$ and $0 \le v \le k$; $\|\cdot\|$ denotes amplitude operation; γ is a hyper-parameter controlling distribution smoothness ($\gamma = 1.5$ is used experientially); $DFT(\cdot)$ denotes discrete Fourier transform. Finally, we sample few trainable spectrum points for complex matrix F of Eq. (3) by following the probability $p(u, v)$.

4. Experiments

4.1. Datasets and Evaluation Protocol

The experiments are conducted on four common tasks. **1) Image Classification:** VTAB [48] and FGVC [10, 23, 32,

Method \ Dataset	Natural							Specialized				Structured								VTAB	
	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Patch Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr/count	Clevr/distance	DMLab	KITTI/distance	dSprites/loc	dSprites/ori	SmallNORB/azi	SmallNORB/ele	Mean Acc.	Mean Params. (%)
Full [22]	68.9	87.7	64.3	97.2	86.9	87.4	38.8	79.7	95.7	84.2	73.9	56.3	58.6	41.7	65.5	57.5	46.7	25.7	29.1	65.5	100.00
Linear [22]	63.4	85.0	64.3	97.0	86.3	36.6	51.0	78.5	87.5	68.6	74.0	34.3	30.6	33.2	55.4	12.5	20.0	9.6	19.2	53.0	0.05
Bias [47]	72.8	87.0	59.2	97.5	85.3	59.9	51.4	78.7	91.6	72.9	69.8	61.5	55.6	32.4	55.9	66.6	40.0	15.7	25.1	62.0	0.16
Adapter [18]	74.1	86.1	63.2	97.7	87.0	34.6	50.8	76.3	88.0	73.1	70.5	45.7	37.4	31.2	53.2	30.3	25.4	13.8	22.1	55.8	0.31
LoRA [19]	68.1	91.4	69.8	99.0	90.5	86.4	53.1	85.1	95.8	84.7	74.2	83.0	66.9	50.4	81.4	80.2	46.6	32.2	41.1	72.6	0.37
LoRA-FA [51]	65.5	89.7	71.2	99.1	90.6	86.7	54.3	83.7	92.6	82.9	74.9	61.9	61.4	43.5	72.3	68.7	45.6	24.2	28.4	68.2	0.17
LoRA+ [14]	69.6	92.5	70.8	98.5	90.4	89.8	53.8	79.9	96.1	86.5	76.5	84.4	65.1	52.8	81.6	78.6	44.2	30.5	40.3	72.7	0.37
HydraLoRA [37]	68.8	92.0	71.3	98.8	90.4	88.0	54.6	84.5	96.0	86.5	74.8	84.8	66.6	53.2	82.0	78.0	46.0	34.5	43.4	73.3	0.56
GPS [53]	70.8	93.9	74.8	99.4	82.4	91.4	51.6	87.2	95.7	86.1	76.1	80.9	61.8	54.0	81.4	84.2	52.6	30.2	45.5	73.6	0.25
SPT-LoRA-8 [15]	72.8	92.7	72.6	99.3	91.2	86.7	55.2	85.5	95.8	85.7	75.6	82.0	68.1	49.4	81.4	80.2	48.6	29.4	39.7	73.3	0.51
FourierFT [9]	67.9	89.7	72.2	99.1	91.0	90.8	55.0	85.5	95.7	86.3	75.6	82.5	68.5	52.9	81.0	75.4	46.0	28.1	38.9	72.8	0.16
S^2FT (Ours)	68.8	91.4	72.6	99.2	91.3	90.7	55.1	86.2	96.1	86.3	76	82.8	69.6	52.7	81.4	79.4	46.8	29.7	41.4	73.6	0.08
S^2FT (Ours)	72.0	91.9	72.5	99.2	91.5	90.6	55.7	86.9	96.2	86.7	76.5	83.8	69.7	53.9	81.7	80.1	48.4	30.5	42.0	**74.1**	0.16

Table 1. Results of image classification on VTAB-1k with ViT-B/16 pre-trained on ImageNet-21K.

Dataset	CUB -2011	NA-Brids	Oxford Flowers	Stan. Dogs	Stan. Cars	Mean Acc.	Params. (%)
Full [22]	87.3	82.7	98.8	89.4	84.5	88.5	100.00
Linear [22]	85.3	75.9	97.9	86.2	51.3	79.3	0.21
Bias [47]	88.4	84.2	98.8	91.2	79.4	88.4	0.33
Adapter [18]	87.1	84.3	98.5	89.8	68.6	85.6	0.48
LoRA [19]	84.9	79.0	98.1	88.1	87.4	87.5	0.34
LoRA-FA [51]	85.8	79.1	99.1	88.1	80.1	86.4	0.17
LoRA+ [14]	86.5	80.8	99.2	87.4	88.7	88.5	0.34
HydraLoRA [37]	86.9	82.1	99.1	88.2	89.1	89.0	0.47
FourierFT [9]	85.5	82.4	99.1	89.0	83.7	87.9	0.16
S^2FT (Ours)	87.9	84.3	99.1	90.6	83.6	89.1	0.08
S^2FT (Ours)	88.4	84.6	99.3	91.4	84.5	**89.6**	0.16

Table 2. Image classification on FGVC

39, 40] datasets. The VTAB comprises 19 distinct visual classification tasks organized into 3 semantic domains, we use data splits following [15]. The FGVC benchmark includes 5 datasets: CUB-200-2011, NABirds, Oxford Flowers, Stanford Cars, and Stanford Dogs, using standardized data splits from [15]. Classification accuracy is used as evaluation protocol. **2) Image Generation:** the subject-driven text-to-image generation task. We use the dataset proposed in [35], where 5 or 6 image samples are used for training each subject. The comparison metrics includes subject fidelity (DINO [2], CLIP-I [34]), text prompt fidelity (CLIP-T [34]), and sample diversity (LPIPS [52]). **3) Natural Language Understanding.** Following [9], we select 6 tasks (i.e., SST-2, MRPC, CoLA, QNLI, RTE, STSB) from the GLUE (General Language Understanding Evaluation[41]) benchmark to compare our S^2FT and baselines, where Acc, MCC, and PCC are used as metrics. **4) Instruction Tuning.** For instruction tuning, we conducted experiments on the Alpaca dataset [42]. During evaluation, following[9] the fine-tuned models are used to answer a set of standardized questions sourced from the MT-Bench [55]and Vicuna [5] Eval benchmark suites. The generated responses are then scored by GPT-4 on a scale from 0 to 10.

4.2. Implementation Details and Baselines

Image Classification. We follow [15] to process the images of FGVC and VTAB-1k. We employ the AdamW optimizer with cosine learning rate decay to fine-tune models for 100 epochs and 10 epochs linear warm-up. The baseline methods include reparameterization-based methods such as LoRA [19], (LoRA-FA[51], LoRA+[14], HydraLoRA[37]), FourierFT[9] and other latest methods. In particular, we sample 6000 points for FourierFT and 3000 points for our S^2FT for parameter efficiency.

Method	DINO ↑	CLIP-I ↑	CLIP-T ↑	LPIPS ↑	Params(%)
RealImages	0.703	0.864	–	0.695	–
DreamBooth	0.614	0.778	0.239	0.737	100
LoRA	0.613	0.765	0.237	0.744	1.44
FourierFT	0.607	0.750	0.237	0.732	0.07
S^2FT(Ours)	**0.620**	**0.784**	0.234	**0.769**	0.06

Table 3. Results of subjective generation task.

Image Generation. We set hyperparameter by following [35] and select latest Dreambooth, LoRA and FourierFT as our baseline. Here, we sample 10000 and 8000 points for FourierFT and our S^2FT.

Natural Language Understanding. Following [9], we compare our S^2FT with wide range of models, including Adapter, LoRA, AdaLoRA, DyLoRA, CorDA, FourierFT. The methods are fine-tuned on the commonly used RoBERTa model, including both base and large version.

Instruction Tuning. Following [9], we apply LoRA, FourierFT and our S^2FT to fine-tune several base models from the LLaMA2 family, including LLaMA2-7B and LLaMA2-13B. Unless otherwise noted, we use the same hyperparameter settings as FourierFT for fair comparison.

4.3. Discussion of Results

Image Classification Tasks. In Tables 1 and 2, we report the performance of our S^2FT and baselines on VTAB and FGVC. It can be seen that 1) compared with recent PEFT methods, our S^2FT achieves superior performance on most datasets, which verifies the effectiveness of our S^2FT; 2) In Fourier transforms-based PEFT method, our S^2FT method achieves a significant improvement (around $1\% \sim 2\%$). In particular,

Model	Params (%)	SST-2 (Acc.)	MRPC (Acc.)	CoLA (MCC)	QNLI (Acc.)	RTE (Acc.)	STS-B (PCC)	Avg.
RoB_{base}(Full)	100.0	94.8	90.2	63.6	92.8	78.7	91.2	85.2
BitFit	0.08	93.7	**92.7**	62	91.8	**81.5**	90.8	85.4
$\text{Adapter}^{\text{D}}$	0.24	94.2	88.5	60.8	93.1	71.5	89.7	83.0
$\text{Adapter}^{\text{D}}$	0.72	94.7	88.4	62.6	93.0	75.9	90.3	84.2
LoRA	0.24	95.1	89.7	63.4	**93.3**	78.4	**91.5**	85.2
AdaLoRA	0.24	94.5	88.7	62.0	93.1	81.0	90.5	85.0
DyLoRA	0.24	94.3	89.5	61.1	92.2	78.7	91.1	84.5
CorDA	0.24	93.1	89.7	59.6	91.5	76.2	90.2	83.4
FourierFT	0.02	94.2	90.0	63.8	92.2	79.1	90.8	85.0
S^2FT	0.02	**95.3**	90.4	**64.8**	92.8	79.8	90.7	**85.6**
$\text{RoB}_{\text{large}}$(Full)	100.0	96.4	**90.9**	68.0	94.7	86.6	**92.4**	88.2
$\text{Adapter}^{\text{P}}$	0.86	96.1	90.2	68.3	**94.8**	83.8	92.1	87.6
$\text{Adapter}^{\text{P}}$	0.22	**96.6**	89.7	67.8	**94.8**	80.1	91.9	86.8
LoRA	0.22	96.2	90.2	68.2	**94.8**	85.2	92.3	87.8
DyLoRA	0.22	94.7	90.7	65.3	93.6	87.2	91.4	87.2
PiSSA	0.22	95.8	91.5	68.1	94.4	87.9	92.0	88.2
FourierFT	0.01	96.0	**90.9**	67.1	94.4	87.4	91.9	88.0
S^2FT	0.01	96.4	90.7	67.6	94.6	**88.1**	92.3	**88.3**

Table 4. GLUE results with RoBERTa

Model	Method	# Trainable Parameters	MT-Bench	Vicuna
LLaMA2-7B	LoRA†	159.9M	$5.19_{\pm.1}$	$7.38_{\pm.3}$
	LoRA	33.5M	$5.20_{\pm.3}$	$7.35_{\pm.6}$
	FourierFT	0.064M	$5.18_{\pm.3}$	$7.49_{\pm.4}$
	S^2FT	0.064M	$\mathbf{5.21}_{\pm.4}$	$\mathbf{7.50}_{\pm.6}$
LLaMA2-13B	LoRA†	250.3M	$5.78_{\pm.2}$	$7.89_{\pm.5}$
	LoRA	52.4M	$5.80_{\pm.2}$	$7.89_{\pm.6}$
	FourierFT	0.08M	$5.82_{\pm.3}$	$7.92_{\pm.5}$
	S^2FT	0.08M	$\mathbf{5.89}_{\pm.4}$	$\mathbf{7.94}_{\pm.6}$

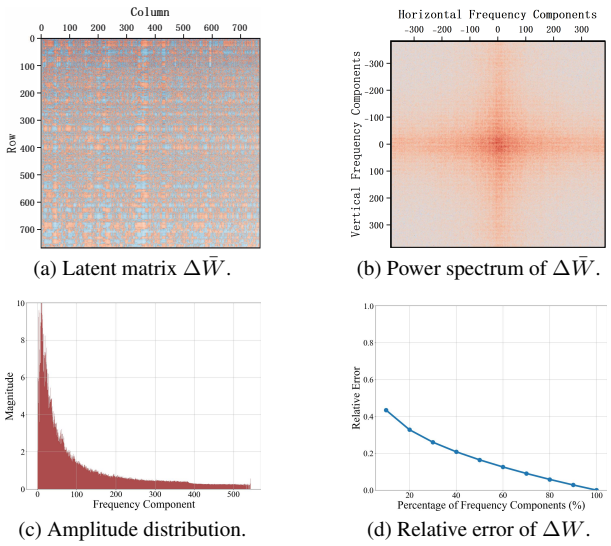
Table 5. The average scores on MT-Bench and Vicuna assessed by GPT-4. † indicates updating the layers other than `lm_head`. Higher score is better.

our method uses only 50% of the parameters compared to FourierFT. This indicates that our method better leverages the expressive power of the spectrum domain.

Image Generation Tasks. From Table 3, we can see that (1) S^2FT significantly outperforms FourierFT on CLIP-I and LPIPS, while achieving slightly better or comparable performance on DINO and CLIP-T; and (2) our method achieves performance comparable to DreamBooth (100%) and LoRA (1.44%) with only 0.06% of the total parameters. The results suggest that our method not only reduces the number of trainable parameters but also yields better generation quality.

Nature Language Understanding Tasks. Table 4 presents the results of natural language understanding. It can be found that our S^2FT achieves superior performance on both the base and large versions of RoBERTa in average with the least trainable parameters scale. Compared with FourierFT, our S^2FT constantly improves 0.6% and 0.3% in average score using the same scale of parameters. Compared with the commonly used LoRA, our S^2FT achieves better result using 12 times smaller parameters. It is worth noting that our S^2FT is the only method that even outperforms full finetuning with 100% parameters in average, demonstrating the versatility of our approach.

Instruction Tuning Tasks. Table 5 presents the results of the instruction tuning task. As shown, our S^2FT consistently outperforms other methods on both LLaMA2-7B and LLaMA2-13B in average performance, while requiring only a minimal number of trainable parameters. In particular, compared to the widely adopted LoRA, S^2FT achieves better performance using merely 0.15% of its parameter count.



(a) Latent matrix $\Delta\bar{W}$. (b) Power spectrum of $\Delta\bar{W}$. (c) Amplitude distribution. (d) Relative error of ΔW.

Figure 5. Distribution analysis of the spatial-domain matrix $\Delta\bar{W}$ obtained by our S^2FT.

4.4. Ablation Study

Impact of number of trainable parameters. Figure 6 reports the accuracy of our S^2FT and baseline on eight tasks from the VTAB dataset using different number of trainable parameters. It can be found that (1) the accuracy of both methods improves as the number of trainable parameters increases; and (2) our S^2FT consistently outperforms baseline, indicating the superiority and efficiency of our S^2FT.

Can our S^2FT seek a latent spatial-domain matrix $\Delta\bar{W}$ with sparse spectrum? To answer this question, in Figure 5, we visualize the spatial-domain matrix $\Delta\bar{W}$ obtained by our invertible transformation $T^{-1}()$, and its power spectrum, amplitude distribution, and relative error of modeling ΔW. Compared with FourierFT (see Figure 2), we find that the spatial-domain matrix $\Delta\bar{W}$ indeed presents smooth property (see Figure 5a) and sparse spectrum (see Figures 5b and 5c), and show superior modeling performance with only a few of spectrum coefficients (see Figure 5d) . This verifies the superiority of our S^2FT on modeling the weight change ΔW with few coefficients.

How does the accuracy of weight change pre-estimation affect performance? In Table 6, we evaluate our S^2FT using different pre-estimation methods, including our method defined in Eq. 4 and the full finetuning (FF) pre-estimation method with different epochs. Note that FF with more optimization epochs yields more accurate pre-estimations, as it better fits the training samples of downstream tasks. Despite

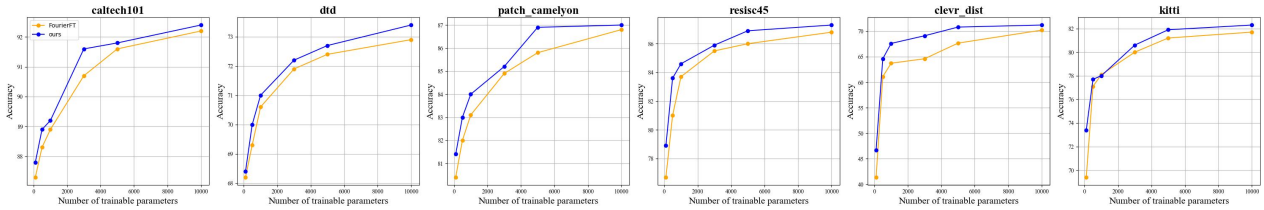


Figure 6. Performance comparison between FourierFT and our S^{2}FT on trainable parameters.

Method	Our Eq. 4	FF (1 epoch)	FF (2 epoch)	FF (3 epoch)	FF (5 epoch)
S^{2}FT	73.6	73.6	73.7	73.7	73.8

Table 6. Impact Analysis of pre-estimation accuracy.

Method	Natural	Specialized	Structured	Mean Acc
S^{2}FT + Random sampling	79.6	84.9	58.9	72.0
S^{2}FT + Sampling toward LF	80.6	85.5	59.7	72.8
S^{2}FT + Our sampling	81.3	86.2	60.5	**73.6**

Table 7. Analysis of sampling method (LF is low frequency).

Method	Training time / epochs(s)	Trans. / Trans.$^{-1}$(s)	Training Memory (GB)	Params. (%)
Full	6.2	-	18.8	100
LoRA	3.8	-	9.7	0.37
FourierFT	4.0	-	8.7	0.16
S^{2}FT	4.2	0.02	8.5	0.08

Table 8. Analysis of training cost on the VTAB dataset.

this, the results show that the accuracy of the weight change pre-estimation has little impact on the performance. This is reasonable, as the pre-estimation is only used to guide us seeking a coarse row/column rearrangement, and the method does not rely heavily on its precision.

Is there a frequency bias in our S^{2}FT? In Figure 7 of appendix, we conduct an experiment using a sampling strategy biased toward different central frequencies to select spectrum points. We can see that our S^{2}FT achieves better performance when favoring lower central frequencies, indicating that the spectrum domain in our method is sparse. Moreover, we observe an interesting but previously unexplained phenomenon noted in the original FourierFT paper [9]: the optimal central frequency for FourierFT varies randomly across tasks, and low frequencies are often not the most effective. This can be attributed to the fact that the spectrum domain in FourierFT is not sparse but rather power-uniform.

Is our spectral coefficient sampling effective? In Table 7, we report the accuracy of our S^{2}FT with different spectrum coefficient sampling strategy on VTAB benchmark. We can see that our proposed sampling strategy consistently performs better with the ones proposed in [1], which verifies the superiority of our sampling strategy.

Is our S^{2}FT computation efficient? In Table 8, we report the training cost comparison of our S^{2}FT, FourierFT, and LoRA using the ViT base model. While consuming similar inference time and memory. Compared with LoRA and FourierFT, S^{2}FT also requires smaller memory cost because our S^{2}FT only consumes smaller training parameter scale.

How does the hyper-parameter γ impact the performance of our S^{2}FT? In Table 9 we report the performance of our

Method	Params. (%)	Natural	Specialized	Structured	Mean Acc
S^{2}FT($\gamma = 0.5$)	0.08	80.7	84.9	58.6	72.3
S^{2}FT($\gamma = 1.0$)	0.08	81.1	84.8	58.9	72.5
S^{2}FT($\gamma = 1.5$)	0.08	81.3	86.2	60.5	73.6
S^{2}FT($\gamma = 2.0$)	0.08	81.0	85.9	60.5	73.4
S^{2}FT($\gamma = 3.0$)	0.08	80.8	85.4	60.2	73.1
S^{2}FT($\gamma = 5.0$)	0.08	80.0	85.1	60.1	72.7

Table 9. Impact analysis on hyper-parameter λ.

S^{2}FT under different values of the hyper-parameter γ on VTAB benchmark, which controls the smoothness of sampling probability. As shown, when γ increases from 0.5 to 1.5, the mean accuracy steadily improves from 72.3% to 73.6%, indicating that a moderate enhancement of smoothness helps better exploit the sparse spectral distribution. The performance peaks at $\gamma = 1.5$, where S^{2}FT achieves the best performance across the three categories (Natural, Specialized, Structured). However, further increasing γ beyond 1.5 leads to a gradual decline in performance, possibly due to over-smoothing, which may be inconsistent with the underlying distribution information. Therefore, $\gamma = 1.5$ is adopted as the default setting in our experiments for its empirical effectiveness.

5. Conclusions

In this paper, we identified a key challenge overlooked in Fourier-based parameter-efficient fine-tuning (PEFT), i.e., the power spectrum of weight change ΔW is not sparse, but tends to uniform. This resulted in that using few spectral coefficients is difficult to accurately model ΔW. To address this challenge, we presented a novel PEFT method with sparse spectrum domain, which aims to seek an invertible transformation that transforms a latent spatial-domain matrix with sparse spectrum to the weight change, and then perform PEFT on such sparse spectrum domain with few spectral coefficients. Results showed that our S^{2}FT achieves superior performance over previous methods.

Acknowledgments

This work was supported by the NSFC under Grant No. 62502120, 62272130 and 62501408, the Guangdong Basic and Applied Basic Research Foundation under Grant No. 2025A1515011674, Shenzhen Science and Technology Program No. ZDCYKCX20250901092700001, SYSPG 20241211173609009, and JCYJ20240813142104006.

References

[1] David Brandwood. *Fourier transforms in radar and signal processing*. Artech House, 2012.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Herve Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[3] Hao Chen, Ran Tao, Han Zhang, Yidong Wang, Xiang Li, Wei Ye, Jindong Wang, Guosheng Hu, and Marios Savvides. Conv-adapter: Exploring parameter efficient transfer learning for convnets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1551–1561, 2024.

[4] Zhuo Chen, Rumen Dangovski, Charlotte Loh, Owen M Dugan, Di Luo, and Marin Soljacic. QuanTA: Efficient high-rank fine-tuning of LLMs with quantum-informed tensor adaptation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[5] Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3): 6, 2023.

[6] Tim OF Conrad, Martin Genzel, Nada Cvetkovic, Niklas Wulkow, Alexander Leichtle, Jan Vybiral, Gitta Kutyniok, and Christof Schütte. Sparse proteomics analysis–a compressed sensing-based approach for feature selection and classification of high-dimensional proteomics mass spectrometry data. *BMC Bioinformatics*, 18:1–20, 2017.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[8] Max Ehrlich and Larry S Davis. Deep residual learning in the jpeg transform domain. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3484–3493, 2019.

[9] Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. Parameter-efficient fine-tuning with discrete fourier transform. *arXiv preprint arXiv:2405.03003*, 2024.

[10] Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, and Li Fei-Fei. Fine-grained car detection for visual census estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[11] Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from jpeg. *Advances in Neural Information Processing Systems*, 31, 2018.

[12] Yangyang Guo, Guangzhi Wang, and Mohan Kankanhalli. Pela: Learning parameter-efficient models with low-rank approximation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15699–15709, 2024.

[13] Cheng Han, Qifan Wang, Yiming Cui, Zhiwen Cao, Wenguan Wang, Siyuan Qi, and Dongfang Liu. E 2 vpt: An effective and efficient approach for visual prompt tuning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17445–17456. IEEE, 2023.

[14] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. In *Forty-first International Conference on Machine Learning*.

[15] Haoyu He, Jianfei Cai, Jing Zhang, Dacheng Tao, and Bohan Zhuang. Sensitivity-aware visual parameter-efficient fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11825–11835, 2023.

[16] Haoze He, Juncheng B Li, Xuan Jiang, and Heather Miller. SMT: Fine-tuning large language models with sparse matrices. In *The Thirteenth International Conference on Learning Representations*, 2025.

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[18] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[19] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[20] Qiushi Huang, Tom Ko, Zhan Zhuang, Lilian Tang, and Yu Zhang. HiRA: Parameter-efficient hadamard high-rank adaptation for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[21] Cor A.J. Hurkens and Gerhard J. Woeginger. On the nearest neighbor rule for the traveling salesman problem. *Operations Research Letters*, 32(1):1–4, 2004.

[22] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022.

[23] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, 2011.

[24] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[25] Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. Vera: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations*.

[26] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient

model tuning. *Advances in Neural Information Processing Systems*, 35:109–123, 2022.

[27] Mengqi Liao, Wei Chen, Junfeng Shen, Shengnan Guo, and Huaiyu Wan. HMoRA: Making LLMs more effective with hierarchical mixture of loRA experts. In *The Thirteenth International Conference on Learning Representations*, 2025.

[28] Yan-Bo Lin, Yi-Lin Sung, Jie Lei, et al. Vision transformers are parameter-efficient audio-visual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2299–2309, 2023.

[29] Shih-yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

[30] Otniel-Bogdan Mercea, Alexey Gritsenko, Cordelia Schmid, and Anurag Arnab. Time-memory-and parameter-efficient visual adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5536–5545, 2024.

[31] Niklas Mevenkamp and Benjamin Berkels. Variational multiphase segmentation using high-dimensional local features. In *WACV*, 2016.

[32] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

[33] Zelin Peng, Zhengqin Xu, Zhilin Zeng, Lingxi Xie, Qi Tian, and Wei Shen. Parameter efficient fine-tuning via cross block orchestration for segment anything model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3743–3752, 2024.

[34] Alec Radford, JongWook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Askell Amanda, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *Cornell University - arXiv,Cornell University - arXiv*, 2021.

[35] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[36] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *International conference on machine learning*, pages 21076–21089. PMLR, 2022.

[37] Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Chengzhong Xu. Hydralora: An asymmetric lora architecture for efficient fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[38] Cheng-Hao Tu, Zheda Mai, and Wei-Lun Chao. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7725–7735, 2023.

[39] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 595–604, 2015.

[40] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[41] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018.

[42] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508, 2023.

[43] Sunghyeon Woo, Sol Namkung, Sunwoo Lee, Inho Jeong, Beomseok Kim, and Dongsuk Jeon. PaCA: Partial connection adaptation for efficient fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[44] Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Reft: Representation finetuning for language models. In *Advances in Neural Information Processing Systems*, pages 63908–63962, 2024.

[45] Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[46] Dongshuo Yin, Yiran Yang, Zhechao Wang, Hongfeng Yu, Kaiwen Wei, and Xian Sun. 1% vs 100%: Parameter-efficient low rank adapter for dense predictions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20116–20126, 2023.

[47] Elad Ben Zaken, Yoav Goldberg, and Shauli 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)*, pages 1–9, 2022.

[48] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

[49] Baoquan Zhang, Huaibin Wang, Chuyao Luo, Xutao Li, Guotao Liang, Yunming Ye, Xiaochen Qi, and Yao He. Codebook transfer with part-of-speech for vector-quantized image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7757–7766, 2024.

[50] Baoquan Zhang, Bingqi Shan, Aoxue Li, Chuyao Luo, Yunming Ye, and Zhenguo Li. Zookt: Task-adaptive knowledge transfer of model zoo for few-shot learning. *Pattern Recognition*, 158:110960, 2025.

[51] Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303*, 2023.

[52] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[53] Zhi Zhang, Qizhe Zhang, Zijun Gao, Renrui Zhang, Ekaterina Shutova, Shiji Zhou, and Shanghang Zhang. Gradient-based parameter selection for efficient fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28566–28577, 2024.

[54] Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. In *Forty-first International Conference on Machine Learning*.

[55] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

[56] Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Saez De Ocariz Borde, Rickard Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. In *International Conference on Learning Representations*, 2024.