WST: Wavelet-Based Multi-scale Tuning for Visual Transfer Learning

Jia Zeng[1], **Lan Huang**[1, 2], **Kangping Wang**[1, 2*]

[1]College of Computer Science and Technology, Jilin University
Changchun 130012, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of the Ministry of Education, Jilin University
Changchun 130012, China
zengjia22@mails.jlu.edu.cn, huanglan@jlu.edu.cn, wangkp@jlu.edu.cn

Abstract

Large-scale pre-trained Vision Transformer (ViT) models have demonstrated remarkable performance on visual tasks but are computationally expensive to transfer to downstream tasks. Parameter-Efficient Fine-Tuning (PEFT) offers a promising transferring approach by updating only a subset of parameters. However, PEFT's effectiveness is hindered by discrepancies between pre-training and downstream tasks in terms of object scale and granularity. Downstream tasks often focus on finer-grained and more specialized recognition, requiring more detailed features. The diversity of feature scales of existing PEFT methods for ViT is limited. To address this, we propose a novel PEFT method named Wavelet-based multi-Scale Tuning (WST), which learns multi-scale features in a simple and efficient way. WST introduces a parallel fine-tuning patch embedding branch with a smaller patch size than the pre-trained model to capture finer-grained features. Furthermore, to handle the computational challenge from the resulting longer token sequence, WST designs wavelet fine-tuning blocks that balance both efficiency and performance. In the block, wavelet transform enables invertible and lossless down-sampling of the longer token sequence, aligning it with that of the backbone, and two lightweight linear mappings are employed to learn task-specific features. This design facilitates efficient multi-scale information exchange between the pre-trained backbone and fine-tuning branch. Extensive experiments on transfer learning demonstrate the promising performance and efficiency of our WST.

Code — https://github.com/ZJia-goo/WST

Introduction

With the rapid expansion of dataset scale and model size, large-scale Vision Transformer (ViT) models have achieved remarkable success in many visual tasks (Dosovitskiy et al. 2020; Dehghani et al. 2023; Kirillov et al. 2023). Large-scale ViTs pre-trained on large datasets (such as ImageNet-21K) can learn general features, so pre-trained models are widely used in transfer learning to improve performance and accelerate convergence. However, transferring these large-scale ViTs to various downstream visual tasks by full-tuning faces challenges due to the heavy load on the number of parameters. Recently, Parameter-Efficient Fine-Tuning (PEFT) has been widely explored as a practical solution. PEFT approaches (Houlsby et al. 2019; Hu et al. 2022; Jia et al. 2022; Sung, Cho, and Bansal 2022; Fu, Zhu, and Wu 2024) aim to efficiently adapt pre-trained ViTs to downstream tasks by updating only a small subset of parameters and freezing other parameters. Many PEFT methods such as Adapter (Houlsby et al. 2019), Low-Rank Adaptation (LoRA) (Hu et al. 2022), Visual Prompt Tuning (VPT) (Jia et al. 2022) and so on, have developed great potential. However, existing PEFT methods apply fine-tuning at the same scale and granularity, disregarding differences in scale and granularity between pre-training and downstream tasks. Many downstream tasks especially such as fine-grained and specialized image recognition tasks require finer features for accurate predictions. PEFT methods need to capture not only general features from pre-trained models to improve generalization, but also detailed features to distinguish more similar and specialized patterns. Therefore, it is necessary to integrate multi-scale information in PEFT, particularly at finer levels.

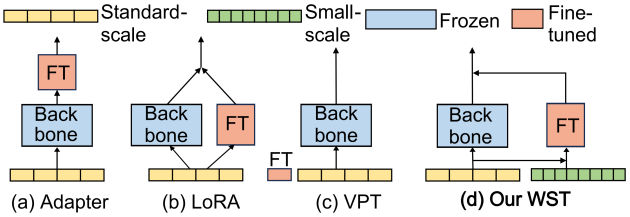

Figure 1: The sketches of previous PEFT methods vs. our WST. Previous methods only consider single-scale features. Our WST integrates a smaller scale (illustrated by green) to learn fine-grained features benefiting for downstream tasks.

The patch size of ViT is an important factor that influences the scale of extracted features. Patch size refers to the size of square sub-sections that an input image is split into. Smaller patches can capture finer image details, leading to richer feature representations and better performance. This finding is consistent with experimental results. For example, ViT-32/B with a patch size of 32 attains 75.9% accuracy on ImageNet-1K while ViT-16/B with a patch size of 16 achieves higher

*Corresponding author

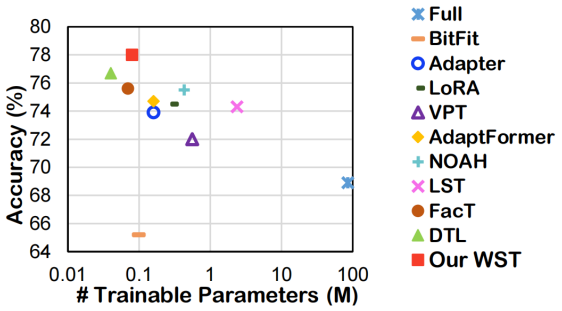

Figure 2: Accuracy and the number of trainable parameters for different methods on VTAB-1K benchmark with ViT-B/16 backbone. Our WST achieves the highest top-1 accuracy with small parameter count compared to other methods.

accuracy at 81.1%. Motivated by this, we attempt to incorporate smaller patch size in addition to the fixed patch size of pre-trained models to enhance performance, as shown in Fig. 1. However, the patch size is inversely proportional to the length of token sequence in ViT. Although smaller patch sizes enhance feature extraction abilities, they significantly increase computational costs, which is a primary challenge for PEFT. Therefore, our main focus in this paper is how to fuse small-scale features to learn more distinguishing information for downstream tasks in an efficient way.

In this paper, we propose a novel PEFT approach named Wavelet-based multi-Scale Tuning (WST). Our WST aims to combine multi-scale features to learn both general and detailed features while maintaining parameter and computational efficiency. First, a fine-tuning small-scale patch embedding branch with smaller patch size than that of pre-trained ViT is introduced in a parallel way. This branch can learn fine-grained task-specific representations. A smaller patch size will result in a longer token sequence, which becomes computationally expensive. To trade off effectiveness and efficiency, WST then designs wavelet-based fine-tuning blocks to handle the longer token sequence and fuse multi-scale features. The blocks consist of a wavelet transform, two lightweight trainable down- and up-projection linear mappings and an inverse wavelet transform. The wavelet transform is applied to realize lossless down-sampling of the token sequence to reduce computational costs. Wavelet transform can decompose features into four sub-bands capturing low and high frequencies, preserving information at all levels. Its invertibility ensures lossless down-sampling and reconstruction. After wavelet transform, the length of token sequence can be reduced and become compatible with that of backbone's token sequence. The two sequences are then added together and fed through two linear layers. The learned features are fused back into the backbone. Finally, token sequence is reconstructed by inverse wavelet transform. WST achieves promising performance and improves parameter efficiency in a simple yet effective way (in Fig. 2). Our main contributions are as follows:

1. We propose a novel PEFT method WST to supplement fine-grained features for downstream tasks. By introducing a small-scale patch embedding branch, WST can enhance the ability to capture detailed task-specific representations.

2. To improve efficiency, WST designs wavelet-based blocks which use wavelet transform for lossless down-sampling over the long token sequence from the small scale branch, reducing the length while preserving information.

3. Extensive experiments show that our WST performs well with very few trainable parameters on various downstream datasets, which outperforms other methods.

Related Work

Parameter-Efficient Fine-Tuning

Recently, with the development of large pre-trained models, there has been a growing interest in PEFT. BitFit (Zaken, Goldberg, and Ravfogel 2022) freezes most parameters and only fine-tunes bias. Adapter (Houlsby et al. 2019) inserts a bottleneck structure composed of two linear layers and a nonlinear activation function after the attention and the Feed-Forward Network (FFN) of ViT. LoRA (Hu et al. 2022) injects trainable low-rank decomposition matrices beside matrices of multi-head self-attention. VPT (Jia et al. 2022) integrates task-specific learnable tokens into inputs. NOAH (Zhang, Zhou, and Liu 2024) searches for optimal module designs, including Adapter, LoRA, and VPT, by neural architecture search. DTL (Fu, Zhu, and Wu 2024) presents a disentangled fine-tuning method, which uses a lightweight side network to disentangle trainable parameters from the model, improving efficiency. However, these methods overlook the importance of feature scale and granularity for downstream tasks, focusing solely on single-scale fine-tuning. By integrating fine scale features into the backbone, our method can capture details besides pre-trained general knowledge, significantly enhancing the model's ability to distinguish subtle patterns crucial for downstream tasks.

Multi-scale Feature Fusion

Image features vary across different scales. Multi-scale feature fusion is actually to sample features of different granularity and fuse them to learn comprehensive representations. CrossViT (Chen, Fan, and Panda 2021) designs a dual-branch architecture which divides the image into large and small patches respectively. CrossViT uses class token of one branch as query to conduct cross attention with the other for scale fusion. SuperViT (Lin et al. 2023) divides the image into patches of different sizes to obtain multi-scale features. SuperViT uses interpolation to up-sample or down-sample patches to align sizes and feeds them into ViT. SSA (Ren et al. 2022) down-samples keys and values to different sizes to capture multi-scale features. In WST, we generate multi-scale features by dividing the image into patches of varying sizes during patch embedding. WST offers distinct advantages. Although CrossViT proposes efficient cross attention, both branches should do self-attention first, which requires high computing resources. SuperViT employs interpolation-based sampling, which will lose information. In contrast, our WST down-samples and aligns sizes via lossless wavelet transform. Additionally, WST learns small-scale features via lightweight linear layers and fuses features via addition, which is simpler and more efficient.

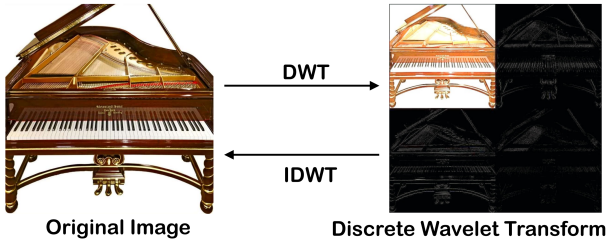

Figure 3: The illustration of DWT and IDWT on images.

Wavelet Transform in Computer Vision

Wavelet transform (Mallat 1989) is a signal analysis method. Wavelet transform is invertible and information-preserving. WaveViT (Yao et al. 2022) uses wavelet transform for down-sampling of key and value to improve computational efficiency. SVT (Patro and Agneeswaran 2024) decomposes features into low-frequency and high-frequency components by wavelet transform and designs a spectral gated network to mix features. Our WST innovatively integrates wavelet transform into PEFT. The efficiency advantage and lossless property of wavelet transform just meet requirements of PEFT for computation and accuracy. Different from WaveViT, our WST down-samples the entire token sequence, leading to a more efficiency gain.

Method

Preliminaries: Wavelet Transform

Wavelet transform is powerful for signal processing. The discrete wavelet transform (DWT) will be used for image processing since image signals are discrete. The DWT can decompose an image into different frequency components and the inverse discrete wavelet transform (IDWT) can reconstruct these frequency components into the original image without loss of information, as shown in Fig. 3. The decomposition ability of DWT comes from low-pass and high-pass filters. The low-pass filter extracts low-frequency information to obtain an approximate image. The high-pass filter allows high-frequency information to pass through to emphasize edges. For simplicity, we choose Haar wavelet. In 2-D DWT, the low-pass filter $f_L = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$ and the high-pass filter $f_H = [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]$ are applied to encode the input $X \in R^{H\times W\times C}$ along rows and columns successively with a stride of 2, resulting in four down-sampled sub-bands: low frequency sub-band $X_{LL} \in R^{\frac{H}{2}\times\frac{W}{2}\times C}$, horizontal high frequency sub-band $X_{LH} \in R^{\frac{H}{2}\times\frac{W}{2}\times C}$, vertical high frequency sub-band $X_{HL} \in R^{\frac{H}{2}\times\frac{W}{2}\times C}$ and diagonal high frequency sub-band $X_{HH} \in R^{\frac{H}{2}\times\frac{W}{2}\times C}$. These sub-bands preserve all details of the input. We can utilize DWT to down-sample features by a factor of 4 without information loss.

Wavelet-Based Multi-scale Tuning

We present an efficient tuning method WST which can see smaller scale to reinforce details for downstream tasks. The architecture of WST is depicted in Fig. 4. We propose two fine-tuning modules: 1) small-scale patch embedding and 2) wavelet fine-tuning block. The two proposed modules can fuse finer-grained features while reducing the computational burden. During fine-tuning, only parameters of the two proposed modules and the head are updated while other parameters remain frozen. The fine-tuning parameters amount occupies a very small proportion, facilitating efficient tuning. We introduce two proposed modules as below.

Small-scale Patch Embedding. Small-scale patch embedding focuses on capturing smaller scale features by employing a smaller patch size. We explain the pipeline briefly. Given an input image $I \in R^{H\times W\times C}$, the image is fed to two distinct embedding branches: 1) the backbone patch embedding with a larger patch size and wider embedding dimension (following standard pre-trained ViT configurations); 2) the small-scale patch embedding with a smaller patch size and smaller embedding dimension. In the backbone patch embedding, the image I is split into non-overlapping patches of size $P \times P \times C$. These patches are linearly embedded by a $P \times P$ convolution with stride P and flattened into token sequence $L^0_{backbone} \in R^{N\times D}$, where $N = \frac{H}{P} \times \frac{W}{P}$ denotes the length and D denotes the dimension.

$$L^0_{backbone} = Flatten(Conv_{P\times P}(I)) \tag{1}$$

In addition, the input image I is also fed into the proposed fine-tuning small-scale patch embedding branch. In this branch, a finer patch size $\frac{P}{2} \times \frac{P}{2} \times C$ is set, resulting in a longer token sequence with the length of $4 \times \frac{H}{P} \times \frac{W}{P}$ which is four times longer than the backbone branch's sequence due to the smaller patch size. To mitigate computational overhead, the embedding dimension in this branch is reduced to $\frac{D}{4}$. Concretely, the image is embedded by a $\frac{P}{2} \times \frac{P}{2}$ convolution with stride $\frac{P}{2}$ and flattened into token sequence $L^0_{finetuning} \in R^{(4\times N)\times\frac{D}{4}}$.

$$L^0_{finetuning} = Flatten(Conv_{\frac{P}{2}\times\frac{P}{2}}(I)) \tag{2}$$

Multi-scale features are extracted by the two branches. Subsequently, the generated token sequences $L^0_{backbone}$ and $L^0_{finetuning}$ are fed into their respective encoder blocks, where exchange information and fuse multi-scale features.

Wavelet Fine-tuning Block. In this block, multi-scale features are fused in an efficient way. Since the token sequence of the small-scale branch is longer, wavelet transform is employed to carry out lossless down-sampling to reduce sequence length. Specifically, in the i-th block, the input sequence $L^i_{finetuning} \in R^{(4\times N)\times\frac{D}{4}}$ is reshaped into a 3D tensor $X^i_{finetuning} \in R^{(2\times\sqrt{N})\times(2\times\sqrt{N})\times\frac{D}{4}}$. Then DWT is applied to $X^i_{finetuning}$, leading to four down-sampling sub-bands: X^i_{ft-LL}, X^i_{ft-LH}, X^i_{ft-HL}, $X^i_{ft-HH} \in R^{\sqrt{N}\times\sqrt{N}\times\frac{D}{4}}$. The four sub-bands are then concatenated along the channel dimension to $X^i_{wave} = [X^i_{ft-LL}, X^i_{ft-LH}, X^i_{ft-HL}, X^i_{ft-HH}] \in R^{\sqrt{N}\times\sqrt{N}\times D}$ and then flattened to $L^i_{wave} \in R^{N\times D}$.

$$X^i_{finetuning} = Reshape(L^i_{finetuning}) \tag{3}$$

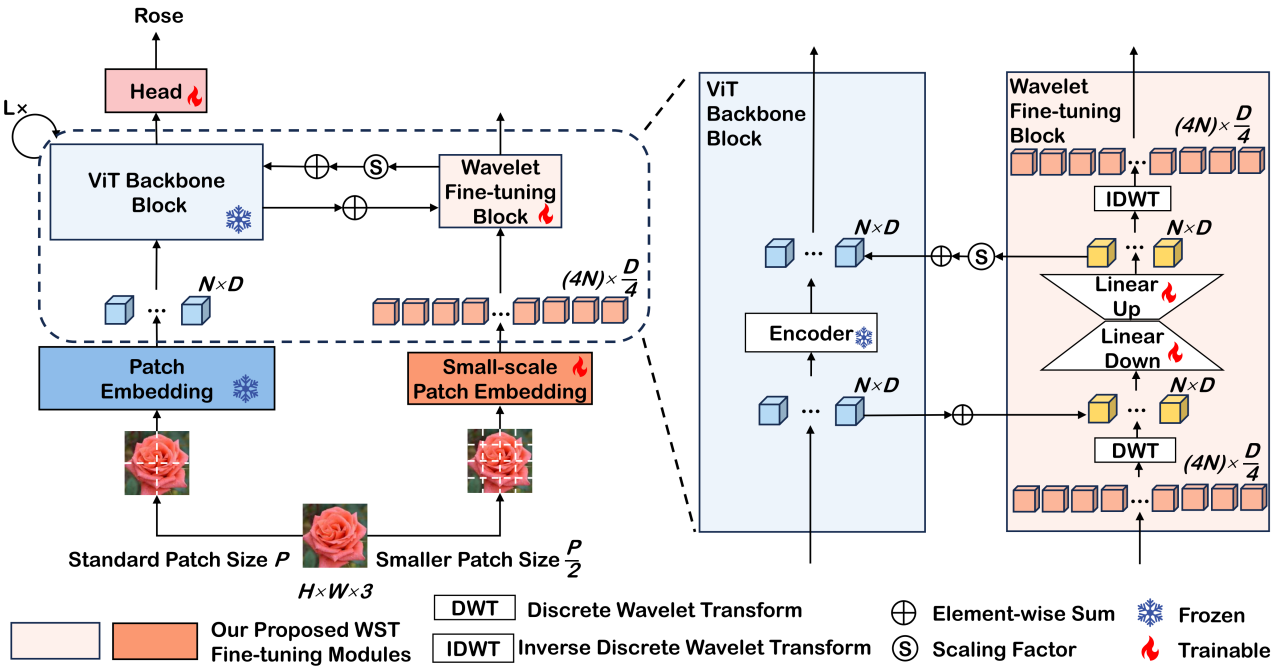

Figure 4: Illustration of our WST on ViT backbone. The Small-scale Patch Embedding module and Wavelet Fine-tuning Blocks are our proposed lightweight fine-tuning modules. In the Small-scale Patch Embedding, a smaller patch size of $\frac{P}{2}$ and a smaller dimension of $\frac{D}{4}$ are set to generate a 4-time longer but with $\frac{1}{4}$ of the dimension compared to the backbone sequence. In Wavelet Fine-tuning Blocks, the longer sequence is down-sampled by DWT, aligning the backbone sequence. Then the two sequences are added together and two low-rank linear layers are applied to learn task-specific features. The features are multiplied by a scaling factor and added back to the backbone to fuse multi-scale features. IDWT is used to reconstruct the sequence.

$$X^i_{wave} = Concat(DWT(X^i_{finetuning})) \tag{4}$$

$$L^i_{wave} = Flatten(X^i_{wave}) \tag{5}$$

In this ingenious way, the length N and dimension D of the token sequence L^i_{wave} of the fine-tuning branch are matched with those of token sequence $L^i_{backbone}$ of the backbone branch. After DWT, a residual-like structure is designed to combine the two branches. The backbone token sequence $L^i_{backbone}$ is added to the down-sampled token sequence L^i_{wave} of the fine-tuning branch. Two lightweight linear layers $W^i_{down} \in R^{D\times r}$ and $W^i_{up} \in R^{r\times D}$, where $r \ll D$, are then applied to learn task-specific features. $L^i_{backbone}$ is also fed into the pre-trained ViT backbone block to learn general features. Finally, the learned features are multiplied by a scaling factor s and are added back to the backbone.

$$L^i = L^i_{wave} + L^i_{backbone} \tag{6}$$

$$\hat{L^i} = (L^i \cdot W^i_{down}) \cdot W^i_{up} \tag{7}$$

$$L^{i+1}_{backbone} = BackboneBlock^i(L^i_{backbone}) + s \cdot \hat{L^i} \tag{8}$$

After feature fusion, inverse wavelet transform IDWT is applied to reconstruct the token sequence back to the original size, leading to $L^{i+1}_{finetuning} \in R^{(4\times N)\times \frac{D}{4}}$.

$$L^{i+1}_{finetuning} = IDWT(\hat{L^i}) \tag{9}$$

Advantages

Rich features & Small parameters amount. Our proposed fine-tuning modules can effectively learn richer, multi-scale features with minimal trainable parameters. In the small-scale patch embedding branch, the number of convolution weight parameters is $\frac{P}{2} \times \frac{P}{2} \times 3 \times \frac{D}{4}$(no consideration of bias). As the patch size $\frac{P}{2}$ and dimension $\frac{D}{4}$ are both small, the number of parameters is very low. More importantly, the patch size and the dimension are carefully designed. The size $(4 \times N) \times \frac{D}{4}$ of resulting token sequence can be precisely compatible with the size $N \times D$ of the backbone's sequence after wavelet transform's lossless down-sampling. In wavelet fine-tuning blocks, the middle dimension r of linear layers is set to 2 or 4, which is far fewer than previous PEFT methods (e.g. 8 is set in Adapter and LoRA). Therefore, WST can ingeniously obtain detailed and informative feature representations while maintaining a small parameter count and low computational overhead, offering dual advantages for PEFT tasks.

Experiments

We conduct a series of experiments to evaluate our WST. 1) We evaluate effectiveness of WST on VTAB-1K benchmark for basic transfer learning tasks. 2) We verify our WST on few-shot learning. 3) We evaluate the generalization ability of WST on domain generalization. 4) We verify our WST

		Natural							Specialized				Structured								
	#params (M)	Cifar100	Caltech101	DTD	Flower102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	Average
Traditional Fine-Tuning Methods																					
Full	85.8	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	68.9
Linear	0	64.4	85.0	63.2	97.0	86.3	36.6	51.0	78.5	87.5	68.5	74.0	34.3	30.6	33.2	55.4	12.5	20.0	9.6	19.2	57.6
Parameter-Efficient Fine-Tuning Methods																					
BitFit	0.10	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	65.2
Adapter	0.16	69.2	90.1	68.0	98.8	89.9	82.8	54.3	84.0	94.9	81.9	75.5	80.9	65.3	48.6	78.3	74.8	48.5	29.9	41.6	73.9
LoRA	0.29	67.1	91.4	69.4	98.8	90.4	85.3	54.0	84.9	95.3	84.4	73.6	82.9	69.2	49.8	78.5	75.7	47.1	31.0	44.0	74.5
VPT	0.56	**78.8**	90.8	65.8	98.0	88.3	78.1	49.6	81.8	96.1	83.4	68.4	68.5	60.0	46.5	72.8	73.6	47.9	32.9	37.8	72.0
AdaptFormer	0.16	70.8	91.2	70.5	99.1	90.9	86.6	54.8	83.0	95.8	84.4	**76.3**	81.9	64.3	49.3	80.3	76.3	45.7	31.7	41.1	74.7
NOAH	0.43	69.6	92.7	70.2	99.1	90.4	86.1	53.7	84.4	95.4	83.9	75.8	82.8	68.9	49.9	81.7	81.8	48.3	32.8	44.2	75.5
FacT	0.07	70.6	90.6	70.8	99.1	90.7	88.6	54.1	84.8	96.2	84.5	75.7	82.6	68.2	49.8	80.7	80.8	47.4	33.2	43.0	75.6
E^2VPT	0.25	78.6	89.4	67.8	98.2	88.5	85.3	52.3	82.5	**96.8**	84.8	73.6	71.7	61.2	47.9	75.8	80.8	48.1	31.7	41.9	73.9
SA^2VP	-	73.0	91.9	70.5	99.1	90.8	84.7	56.8	86.0	95.9	85.8	75.2	76.6	61.8	50.8	79.9	84.5	52.8	**34.7**	45.3	75.8
Hydra	0.28	72.7	91.3	72.0	99.2	91.4	90.7	55.5	85.8	96.0	86.1	75.9	**83.2**	68.2	50.9	**82.3**	80.3	50.8	34.5	43.1	76.5
DTL	0.04	69.6	**94.8**	71.3	99.3	91.3	83.3	56.2	87.1	96.2	86.1	75.0	82.8	64.2	48.8	81.9	**93.9**	53.9	34.2	47.1	76.7
WST	0.08	75.5	92.7	**74.5**	**99.4**	**91.8**	**90.9**	**57.8**	**87.3**	96.4	**87.8**	74.8	82.3	**71.8**	**52.4**	81.7	88.3	**54.9**	32.8	**48.2**	**78.0**

Table 1: Top-1 accuracy on VTAB-1K benchmark with ViT-B/16 backbone. "#params (M)" indicates the number of tunable parameters. "Average" represents the mean group-wise top-1 accuracy across three groups. Best results are in bold.

Method	#p (M)	Nat.	Spe.	Str.	Avg.
Full	86.7	79.2	86.2	59.7	75.0
Linear	0	73.5	80.8	33.5	62.6
BitFit	0.20	74.2	80.1	42.4	65.6
VPT	0.16	76.8	84.5	53.4	71.6
FacT	0.14	**83.1**	86.9	62.1	77.4
DTL	0.09	82.4	87.0	64.2	77.9
WST	0.11	83.0	**87.2**	**64.5**	**78.2**

Table 2: Results on VTAB-1K benchmark with Swin-B backbone. '#p (M)' indicates the number of trainable parameters. 'Nat.'/'Spe.'/'Str.'/'Avg.' denote the average top-1 accuracy for 'Natural', 'Specialized' and 'Structured' group and three groups of VTAB-1K. Best results are in bold.

on fine-grained classification. 5) We conduct ablation experiments and visualization to analyze our method. During training, backbone's parameters remain frozen and only the parameters of the proposed small-scale patch embedding, wavelet fine-tuning blocks and head are fine-tuned.

Experiments on VTAB-1K

Dataset. VTAB-1K benchmark (Zhai et al. 2019) is a classical benchmark to evaluate the transfer ability. It contains 19 vision datasets categorized into 3 groups: 1) Natural group, including generic and fine-grained objects; 2) Specialized group, containing images captured by specialist equipment, such as medical or remote sensing images; 3) Structured group designed for scene structure comprehension, such as depth prediction, object counting and orientation detection. Each dataset contains 1,000 images for training. We report top-1 accuracy on the test set.

Settings. We choose ViT-B/16 (Dosovitskiy et al. 2020) and Swin-B (Liu et al. 2021) pre-trained on ImageNet-21K (Deng et al. 2009) as backbones. For ViT-B/16, the patch size P is 16 and embedding dimension D is 768. Therefore, the patch size of the small-scale patch embedding branch is $\frac{P}{2} = 8$, and the dimension is $\frac{D}{4} = 192$. The middle dimension r in two linear layers is set to 2. The model is fine-tuned for 100 epochs. The AdamW optimizer (Loshchilov and Hutter 2019) is employed. The learning rate schedule adopts cosine decay strategy with decay=0.05 and 10-epoch of linear warm-up. Images are resized to 224×224. We only adopt standard augmentation strategies without MixUp (Zhang et al. 2018) and CutMix (Yun et al. 2019). The convolution weights of small-scale patch embedding are initialized with a Kaiming-uniform (He et al. 2015) distribution. The weights and bias of two linear layers apply trunc-normal initialization and zero initialization.

In Swin-B, patches are merged across stages due to its hierarchical architecture. Therefore, the patch size and the dimension are doubled stage by stage. In the fine-tuning branch, the patch size of small-scale patch embedding keeps $\frac{1}{2}$ that of the backbone branch and the dimension keeps $\frac{1}{4}$ that of the backbone branch. The middle dimension r in two linear mappings is set to 4. Other experimental settings are similar to those of ViT-B/16.

Baselines. We choose two kinds of fine-tuning methods for comparisons. For traditional methods, 'Full' represents full fine-tuning and 'Linear' means that only the classifica-

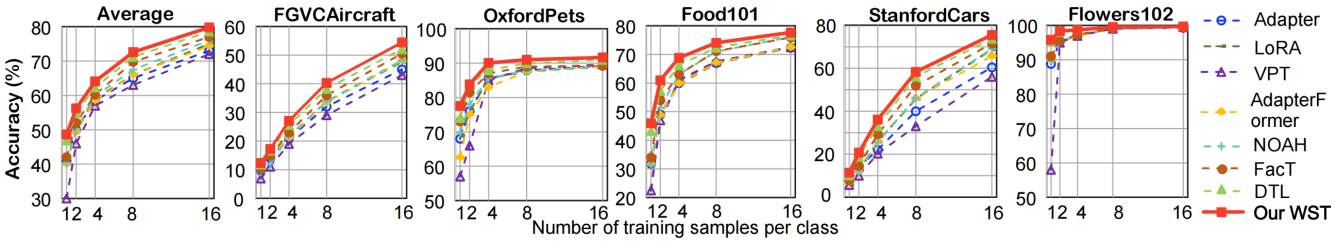

Figure 5: Top-1 accuracy on five fine-grained few-shot learning benchmarks with ViT-B/16 backbone.

Method	Source	Target			
	ImageNet	-Sketch	-V2	-A	-R
Adapter	70.5	16.4	59.1	5.5	22.1
VPT	70.5	18.3	58.0	4.6	23.2
LoRA	70.8	20.0	59.3	6.9	23.3
NOAH	71.5	24.8	66.1	11.9	28.5
DTL	78.3	35.4	67.8	14.0	34.4
WST	**78.8**	**37.7**	**68.4**	**15.6**	**36.8**

Table 3: Top-1 accuracy on domain generalization with ViT-B/16 backbone. Best results are in bold.

tion layer is fine-tuned. For PEFT, we select recent methods including: BitFit (Zaken, Goldberg, and Ravfogel 2022), Adapter (Houlsby et al. 2019), LoRA (Hu et al. 2022), VPT (Jia et al. 2022), AdaptFormer (Chen et al. 2022), NOAH (Zhang, Zhou, and Liu 2024), FacT (Jie and Deng 2023), E^2VPT (Han et al. 2023), SA^2VP (Pei et al. 2024), Hydra (Kim et al. 2024) and DTL (Fu, Zhu, and Wu 2024).

Results. We mainly conduct fine-tuning experiments on ViT/B-16, with results summarized in Table 1. Our WST achieves an average accuracy of 78% while requiring only 0.08M trainable parameters, surpassing previous state-of-the-art (SOTA) methods by 1.3%. Notably, our WST achieves SOTA performance on 11 out of 19 tasks. WST can address object size and granularity mismatches between pre-trained tasks and downstream tasks by efficiently incorporating fine-scale features. Our WST excels in fine-grained and specialized tasks demanding precise recognition based on detailed features. Specifically, in some fine-grained tasks on VTAB-1K such as Flower102 (flower classification) and DTD (texture classification), and in specialized tasks such as Resisc45 for remote sensing scene classification, our WST achieves advanced accuracy. Furthermore, the number of trainable parameters introduced by our WST only accounts for 0.09% of total parameters, highlighting the high efficiency of our wavelet transform strategy.

Our WST is also applicable to Swin-B backbones. In Table 2, WST outperforms other methods. Our model has achieved good performance not only for natural groups with small domain gap with pre-trained tasks but also for structured groups with large domain gap, indicating that our tuning strategy are effective for different kinds of datasets.

Method	#params (M)	Avg.
VPT	0.56	89.1
E^2VPT	0.25	89.2
SA^2VP	-	90.1
SSF	0.39	90.7
WST	0.08	**90.7**

Table 4: Top-1 accuracy on fine-grained visual classification with ViT-B/16 backbone. Best results are in bold.

Experiments on Few-Shot Learning

Datasets and Settings. Few-shot learning is a challenge scenario where there are only a few training samples per task. The benchmark in the few-shot experiments are five fine-grained datasets, namely: Aircraft (Maji et al. 2013), Pets (Parkhi et al. 2012), Food-101 (Bossard, Guillaumin, and Van Gool 2014), Cars (Krause et al. 2013) and Flowers102 (Nilsback and Zisserman 2008). We fine-tune pre-trained ViT-B/16 on few-shot learning. The training set contains {1,2,4,8,16}-shot samples per class (Zhang, Zhou, and Liu 2024; Fu, Zhu, and Wu 2024). We report the average top-1 accuracy on the test set over 3 random seeds.

Results. Results on few-shot learning are shown in the Fig. 5. Our WST achieves better performance than other fine-tuning methods on five datasets across various training sample shots, demonstrating the effectiveness of our WST in the few-shot scenario. For example, on the Cars dataset, our model consistently outperforms previous methods by approximately 2% across different shot settings. On the Pets dataset, while performance is relatively similar across methods, our WST still demonstrates slight improvements.

Experiments on Domain Generalization

Datasets and Settings. We carry out domain generalization experiments to verify WST's robustness under domain shifts. ImageNet-1K is set as the source domain. The training set is constructed by randomly selecting 16 training samples from each class. We evaluate our WST on both the source domain and four target domains. The four target domains include: 1) ImageNet-Sketch (Wang et al. 2019) consisting of sketch-like images with the same class as ImageNet; 2) ImageNet-V2 (Recht et al. 2019) collected from a larger source using the same collection process as ImageNet; 3) ImageNet-A (Hendrycks et al. 2021b) containing natural adversarial images and 4) ImageNet-R (Hendrycks et al.

r	#params (M)	Avg.
1	0.06	76.4
2	0.08	78.0
4	0.12	78.1

Table 5: Ablation results on different middle dimension r.

patch size	Avg.
single ViT-B/8 branch	76.3
single ViT-B/16 branch	75.5
WST	78.0

Table 6: Ablation results on different patches.

2021a) composed of a variety of artistic renditions of ImageNet. We use ViT-B/16 as the pre-trained backbone. Other experimental settings are the same as for few-shot learning. We report the average top-1 accuracy across 3 random seeds.

Results. As shown in Table 3, our WST achieves better accuracy on ImageNet than other methods. On the four domain generalization datasets, our WST also outperforms previous SOTA methods. Specifically, on the ImageNet-Sketch and ImageNet-R, our WST has improved accuracy by about 2% compared to the DTL approach. It demonstrates that our WST exhibits strong robustness in the face of domain shifts.

Experiments on Fine-Grained Visual Classification

Following SSF (Lian et al. 2022), we evaluate our WST on fine-grained visual classification (FGVC) benchmarks, including CUB-200-2011 (Wah et al. 2011), NABirds (Van Horn et al. 2015), Oxford Flowers (Nilsback and Zisserman 2008), Stanford Dogs (Khosla et al. 2011) and Stanford Cars (Krause et al. 2013). The results are shown in Table 4. Our WST achieves good performance on FGVC while using a small number of parameters.

Ablation Studies

Middle Dimension r. We design the middle dimension ablation experiment by changing the middle dimension. We report top-1 accuracy on VTAB-1K using ViT-B/16 backbone in Table 5. The accuracy of model with middle dimension of 1 is lower than that of our WST with middle dimension of 2, despite having a smaller parameter count. Increasing the middle dimension to 4 brings a slight accuracy improvement while the number of parameters is increased. Therefore, the middle dimension r of 2 is the optimal choice for the good trade-off between performance and efficiency.

Small-patch Branch We use a single patch embedding branch with ViT-B/8 or ViT-B/16 backbone to evaluate the value of small-patch size fusion. In Table 6, smaller patch size can achieve higher accuracy. The superior accuracy of multi-scale fusion emphasizes the impact of WST.

Down-sampling strategies. In WST, wavelet transform is employed to efficiently down-sample the longer token sequence due to its invertible and lossless nature. In this experiment, we explore alternative down-sampling strategies

Down-sampling Strategy	#params (M)	Avg.
direct reshaping	0.08	76.6
interpolation down-sampling	0.19	77.2
wavelet down-sampling	0.08	78.0

Table 7: Ablation results on down-sampling strategies.

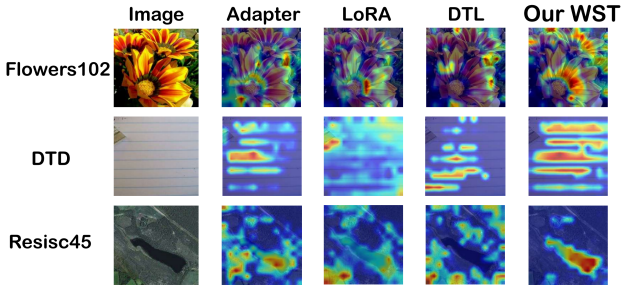

Figure 6: Visualization of saliency maps.

to analyze the effectiveness of wavelet transform. In Table 7, the accuracies of direct reshaping and bilinear interpolation method are lower compared to our wavelet transform approach. Our WST which effectively preserves complete information can enhance model performance.

Visualization

To further show the visual representations, we utilize Grad-CAM (Selvaraju et al. 2017) to visualize saliency maps, highlighting important areas the model focuses on. We choose two fine-grained and one specialized tasks which require more detailed features. In Fig. 6, our WST can better concentrate on key details and semantic information relevant to recognition compared to other fine-tuning methods. It confirms our WST's superior ability to capture fine-grained features, attributed to our small-scale fusion strategy.

Conclusion and Limitation

In this paper, we present Wavelet-based multi-Scale Tuning (WST), a novel paradigm for efficient adapting large-scale pre-trained models to downstream tasks. Due to the intricate nature of downstream tasks requiring a finer level of details, we devote to fusing small-scale features. We introduce an additional parallel small-scale patch embedding branch with a smaller patch size alongside the pre-trained backbone. This enables to focus on a smaller patch receptive field, thereby learning fine-grained features and capturing details. In order to balance the computation and performance, we employ wavelet transform for lossless down-sampling of the token sequence. This facilitates matching token sequence sizes and enables the efficient fusion of multi-scale features. Extensive experiments show that WST can achieve competitive or better performance on various downstream datasets with small trainable parameters count compared to other methods.

A limitation of our method is that the feature size of the small-scale branch is fixed. In future work, we will explore more adaptive incorporation of variable scales through multi-level wavelet transform and neural architecture search.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.62072212), the Jilin Provincial Scientific and Technological Development Program (No.20230201065GX, 20240101364JC) and the Jilin Provincial Key Laboratory of Big Data Intelligent Cognition (No.YDZJ202402075CXJD).

References

Bossard, L.; Guillaumin, M.; and Van Gool, L. 2014. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, 446–461. Springer.

Chen, C.-F. R.; Fan, Q.; and Panda, R. 2021. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, 357–366.

Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; and Luo, P. 2022. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35: 16664–16678.

Dehghani, M.; Djolonga, J.; Mustafa, B.; Padlewski, P.; Heek, J.; Gilmer, J.; Steiner, A. P.; Caron, M.; Geirhos, R.; Alabdulmohsin, I.; et al. 2023. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, 7480–7512. PMLR.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Fu, M.; Zhu, K.; and Wu, J. 2024. Dtl: Disentangled transfer learning for visual recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12082–12090.

Han, C.; Wang, Q.; Cui, Y.; Cao, Z.; Wang, W.; Qi, S.; and Liu, D. 2023. E 2 VPT: An Effective and Efficient Approach for Visual Prompt Tuning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 17445–17456. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; et al. 2021a. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, 8340–8349.

Hendrycks, D.; Zhao, K.; Basart, S.; Steinhardt, J.; and Song, D. 2021b. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15262–15271.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International conference on machine learning*, 2790–2799. PMLR.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 1–13.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual prompt tuning. In *European Conference on Computer Vision*, 709–727. Springer.

Jie, S.; and Deng, Z.-H. 2023. Fact: Factor-tuning for lightweight adaptation on vision transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 1060–1068.

Khosla, A.; Jayadevaprakash, N.; Yao, B.; and Li, F.-F. 2011. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, volume 2.

Kim, S.; Yang, H.; Kim, Y.; Hong, Y.; and Park, E. 2024. Hydra: Multi-head low-rank adaptation for parameter efficient fine-tuning. *Neural Networks*, 106414.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4015–4026.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 554–561.

Lian, D.; Zhou, D.; Feng, J.; and Wang, X. 2022. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35: 109–123.

Lin, M.; Chen, M.; Zhang, Y.; Shen, C.; Ji, R.; and Cao, L. 2023. Super vision transformer. *International Journal of Computer Vision*, 131(12): 3136–3151.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; and Vedaldi, A. 2013. Fine-grained visual classification of aircraft. arXiv:1306.5151.

Mallat, S. G. 1989. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7): 674–693.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, 722–729. IEEE.

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, 3498–3505. IEEE.

Patro, B.; and Agneeswaran, V. 2024. Scattering vision transformer: Spectral mixing matters. *Advances in Neural Information Processing Systems*, 36.

Pei, W.; Xia, T.; Chen, F.; Li, J.; Tian, J.; and Lu, G. 2024. SA^{2}VP: Spatially Aligned-and-Adapted Visual Prompt. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 4450–4458.

Recht, B.; Roelofs, R.; Schmidt, L.; and Shankar, V. 2019. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, 5389–5400. PMLR.

Ren, S.; Zhou, D.; He, S.; Feng, J.; and Wang, X. 2022. Shunted self-attention via multi-scale token aggregation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10853–10862.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Sung, Y.-L.; Cho, J.; and Bansal, M. 2022. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. *Advances in Neural Information Processing Systems*, 35: 12991–13005.

Van Horn, G.; Branson, S.; Farrell, R.; Haber, S.; Barry, J.; Ipeirotis, P.; Perona, P.; and Belongie, S. 2015. 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*, 595–604.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset. California Institute of Technology.

Wang, H.; Ge, S.; Lipton, Z.; and Xing, E. P. 2019. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32.

Yao, T.; Pan, Y.; Li, Y.; Ngo, C.-W.; and Mei, T. 2022. Wave-vit: Unifying wavelet and transformers for visual representation learning. In *European Conference on Computer Vision*, 328–345. Springer.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6023–6032.

Zaken, E. B.; Goldberg, Y.; and Ravfogel, S. 2022. 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)*, 1–9.

Zhai, X.; Puigcerver, J.; Kolesnikov, A.; Ruyssen, P.; Riquelme, C.; Lucic, M.; Djolonga, J.; Pinto, A. S.; Neumann, M.; Dosovitskiy, A.; et al. 2019. A large-scale study of representation learning with the visual task adaptation benchmark. arXiv:1910.04867.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Zhang, Y.; Zhou, K.; and Liu, Z. 2024. Neural prompt search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.