IMPORTANT CHANNEL TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large vision transformers (ViT) have tremendously succeeded in various computer vision tasks. These ViT models pre-trained on large datasets such as ImageNet21K and JFT-300M enjoy robustness in both low-level and high-level visual representations, and they repeatedly yield performance improvements on multiple downstream tasks. One straightforward way to inherit these robust representations is full fine-tuning. However, full fine-tuning is prone to overfitting the small downstream data by adjusting the massive weights of pre-trained large models. In addition, updating the whole parameters of pre-trained large models requires high GPU memory and computations, which limits the application of these large models. To address the above two drawbacks of full fine-tuning, in this paper, we propose a parameter-efficient tuning (PET) method dubbed Important Channel Tuning (ICT). Different from previous PET methods that adopt a trainable module to tune all the channels of a feature map, we hypothesize and corroborate experimentally that not all channels are equal for adaptation. Specifically, we design a tiny external module that determines the most informative channels in the feature map for effective adaptation. In particular, with only a simple linear layer applied to the important channels, our ICT surpasses full fine-tuning on 18 out of 19 datasets in VTAB-1K benchmark by adding only 0.11M parameters of the ViT-B, which is 0.13% of its full fine-tuning counterpart. Moreover, compared with the previous PET methods, ICT achieves the state-of-the-art average performance in the VTAB-1K benchmark with ViT and Swin Transformer backbones.

1 INTRODUCTION

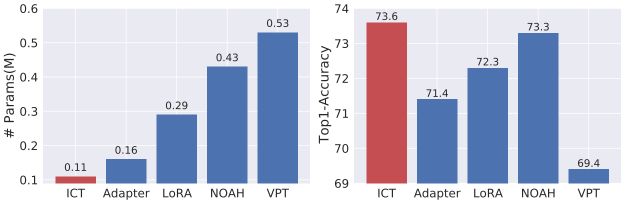


Figure 1: The comparison of parameters and top-1 accuracy on VTAB-1K benchmark with different baselines. The backbone is ViT-B/16.

Large vision transformers (ViT) have shown promising performances on various computer vision tasks such as image classification (Dosovitskiy et al., 2020; Liu et al., 2021b; Yuan et al., 2022; Zhou et al., 2021a), segmentation (Strudel et al., 2021), and detection (Carion et al., 2020; Li et al., 2021). Training large ViT models demands large-scale training data such as ImageNet21K (Ridnik et al., 2021) and JFT-300M (Sun et al., 2017) to fully meet its capacity for strong representations. The pre-trained large ViT models enjoy rich, robust visual representations and can be leveraged in various vision tasks to improve performances. End-to-end full fine-tuning is one direct and commonly used way to inherit these robust representations. However, there are two challenges to adapting these models to the downstream tasks with full fine-tuning. One challenge is that full fine-tuning is prone to overfitting due to the small amount of downstream training data by tuning the massive weights of pre-trained models. The other challenge is the large model size of ViT models, which will cost vast storage and computing resources to save the weights and gradients during fine-tuning. Thus it is unfeasible to tune the large models for a downstream task in resource-limited situations. To mitigate the above two challenges, it is proposed to tune a subset of full parameters or adopt an external trainable module to preserve the plentiful knowledge of pre-trained models and save the tuning cost.

As for tuning a subset of full parameters, there are two common methods: tuning the classification head (Mahajan et al., 2018; Jia et al., 2021; Chen et al., 2021b) and bias term (Cai et al., 2020). Tuning the classification head means freezing the weights of the backbone network and only updating the classification head, which leads to lower performance compared with full fine-tuning. In contrast, tuning the bias term on VTAB-1K benchmark (Zhai et al., 2019) yields surprising results compared with the full fine-tuning as the results provided from Jia et al. (2022). Notably, the proportion of the trainable parameter about the tuning bias term is only 0.10% of the ViT-B/16. This inspires us that the number of parameters of the trainable module could be pretty small.

As for adopting an external trainable module, two lines of parameter-efficient tuning (PET) methods are proposed to preserve the representations of backbones. On the one hand, VPT (Jia et al., 2022) is a visual prompt approach that explores the potential of only leveraging the dataset-specific prompts in the transformer layers, motivated by the success of proposing prompts in natural language processing (Liu et al., 2021a). However, one drawback of VPT is that it utilizes dataset-specific information by individually searching the prompt length per task. Therefore, it is not flexible in applying a new task, as it costs too much time and computation to search for the best prompt length. Another direction (Chen et al., 2022b; Houlsby et al., 2019; Jie & Deng, 2022) is to inject a residual module alongside the multi-head self-attention ("Attn") or "MLP" block in ViT (Dosovitskiy et al., 2020) to adapt the knowledge of the fixed backbone. Regardless the structure of the residual module is MLP-like (Houlsby et al., 2019; Chen et al., 2022b) or Convolution-like (Jie & Deng, 2022), these modules grasp the dataset-specific information implicitly during the fully supervised tuning and consider all the channels equally in an extracted feature map. Therefore, the number of trainable parameters is not quite small because of the full channels projection.

In this paper, we propose Important Channel Tuning (ICT), which explicitly considers the channel inequality of the extracted feature map for different datasets. Specifically, we first present an observation investigating the statistics of the extracted feature maps from the frozen backbone and then design an importance score to reflect the importance of each channel. We choose the top-K largest value of importance score vectors, and later we will only update features of these channels with high important scores. This design enables us to tune only a subset of channels instead of all, inheriting the strong representations of pre-trained backbones for downstream tasks. For different downstream datasets, the selected important channels in each layer are different and demonstrate that explicitly involving dataset-specific information is necessary. For the module implementation, we leverage a linear layer to project the selected important channels into new transformed features, add the residual connection and insert back into the original feature map. Note that, unlike previous PET baselines, which have computational cost in searching for the best prompt length or the best architecture for the knowledge adaptation, our dataset-specific channel selection is conducted before downstream training, and no more repeated channel selection computational cost. The experiments demonstrate that our method surpasses other PET baselines on the VATB-1K benchmark with only 0.11M parameters of ViT-B/16, as shown in Fig. 1. We also evaluate the strategies for selecting important channels and the effectiveness of adopting class-aware information in calculating the importance score. Finally, we also apply ICT to domain generalization task to investigate robustness.

2 Related Work

2.1 Vision Transformers

Transformers (Vaswani et al., 2017) have demonstrated outstanding results on natural language processing and computer vision tasks. Lots of vision transformers (Chen et al., 2021a; d'Ascoli et al., 2021; Dong et al., 2022; Ali et al., 2021; Fan et al., 2021; Han et al., 2021; Rao et al., 2021; Yuan et al., 2021; Touvron et al., 2021; Liu et al., 2021b; Wang et al., 2021; Zhou et al., 2021a) are proposed after the pioneering work ViT (Dosovitskiy et al., 2020). Many of them increase the model size gradually for the state-of-the-art results and learn the rich representations by various architectural designs. Noteworthy, most of them are trained on the natural dataset and have the strong potential to be transferred to other domains/tasks. Adopting these models to the downstream tasks alleviates the training difficulty obviously and achieves promising results rapidly.

2.2 Parameter-efficient Tuning Methods

Parameter-efficient tuning focuses on adopting a trainable module with a small number of parameters for fine-tuning. Two lines of PET have been proposed recently to implicitly involve the dataset-specific information in the model adaptation. On one hand, applying prompts (Jia et al., 2022; Liu et al., 2022; Xing et al., 2022; Zheng et al., 2022; Nie et al., 2022; Wang et al., 2022) to the backbone networks show the success on several vision tasks. On the other hand, adding a residual module (Houlsby et al., 2019; Chen et al., 2022b; Jie & Deng, 2022; Chen et al., 2022a) in the backbone networks also acquires promising results for the balance of performance and effectiveness. Adapter (Houlsby et al., 2019) proposes an MLP-like module with two fully connected layers inserted into the backbone networks. The Adapter gives a successful design of first projecting the original dimensional features into a smaller dimension with one nonlinear layer, and projecting back to the original dimensions. It vastly reduces the number of parameters. Inspired by this design, typically, finding a small number of informative channels in a feature map might be enough for the adaptation. Unlike injecting trainable modules into the transformer blocks, LoRA (Hu et al., 2021) optimizes a low-rank decomposition matrix with a low intrinsic dimension to project the matrices of query, key, and value used in multiheaded self-attention in ViT. As for adopting prompts, VPT injects the prompts into each transformer layer's input space with a small number of extra parameters. However, VPT needs to search the prompt length for each downstream task which takes a long time to tune. In our work, to explicitly reduce the computations of incorporating the dataset-specific information into model adaptation, we design an importance score to determine the important channels before downtream tuning. As for NOAH (Zhang et al., 2022), a neural architecture search algorithm, incorporates Adapter, LoRA, and VPT into its network search space. NOAH brings a strong baseline for performing consistently well on different datasets.

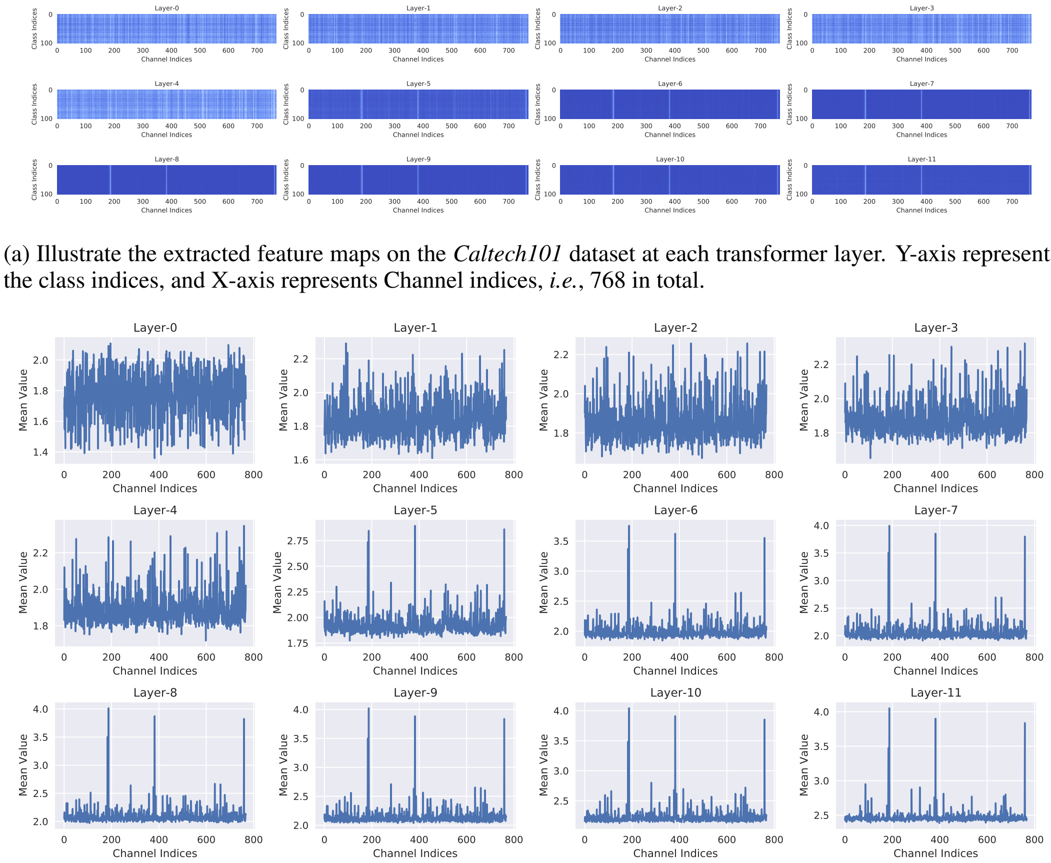
3 Method

3.1 Not all channels are equal

Previous works (Liu et al., 2018; Han et al., 2015; Li et al., 2017) demonstrate that pruning some channels of deep neural networks has a marginal influence on the model performance but can significantly reduce the parameter number and computational cost. Such results reflect that the importance of different channels is not the same, *i.e.*, "Not all channels are equal". Intuitively, the channel importance is different in terms of the datasets, which motivates us to investigate the impact of channel selection in model tuning.

We first illustrate the observation between the pre-trained model and the downstream task. We choose *Caltech101* (Fei-Fei et al., 2004) (one of the downstream tasks from the VTAB-1K benchmark) as an example to evaluate the intermediate features of each transformer layer of ViT. More results of other datasets can be discovered from Fig. 27 to Fig. 45 in the appendix. We choose ViT-B pre-trained on ImageNet21K as the pre-trained model and pass all the training images to extract the features between "Attn" and "MLP" blocks in all 12 transformer layers. As *Caltech101* contains 101 classes and 1000 train images, we divide the total training images into subgroups via the class label and get the deep features $\{f^l_{N_c} \in \mathbb{R}^{B_{N_c}\times L\times C}|\, l, N_c \in N, 1 \le l \le 12 \; and \; 1 \le N_c \le 101\}$ where $\sum_{N_c=1}^{101} B_{N_c} = 1000$, L is the number of tokens, and C is the number of channels. To remove the effects of image bias, we use $L2$ norm to calculate the value of $f^l_{N_c}$ in dimension B_{N_c} and L, then we get features $\{\tilde{f}^l_{N_c} \in \mathbb{R}^{1\times C}|\, l, N_c \in N, 1 \le l \le 12 \; and \; 1 \le N_c \le 101\}$. Fig. 2 (a) shows that for each layer, the mean value of a few channels is much higher than others, *e.g.*, channel indices 183, 382, 636, and 759 in Layer-7. It indicates that when using a pre-trained model to extract the deep features on the target dataset, some channels are more important than others, regardless of categories.

It naturally raises a question: **Could we find important channels in each layer based on deep features extracted from the pre-trained model and then only transform these significant features on downstream tasks?** To answer this question, we propose a criteria to inspect the channel importance of extracted features based on a specific dataset as the guidance to design the trainable module to adapt the information from a pre-trained large model to the target task. Our results in Fig. 1 reflect that only adapting 96 channels of 768 channels could obtain competitive results compared



(a) Illustrate the extracted feature maps on the *Caltech101* dataset at each transformer layer. Y-axis represents the class indices, and X-axis represents Channel indices, *i.e.*, 768 in total.

(b) The plot curves of the important scores are calculated on the *Caltech101* dataset at each transformer layer *i.e.*, Layer-0, Layer-1. Y-axis is the mean value of all classes as shown in (a), and we use log10 for a better display.

Figure 2: The illustration of feature maps and the curves of important scores. All the results are obtained by ViT-B/16 pre-trained on ImageNet21K.

to other baselines. Simultaneously, partially transforming the channels in a feature map reduces the number of learnable parameters heavily (e.g., $12 \times 96 \times 96$ is quite smaller than the $12 \times 768 \times 768$).

3.2 Class-Aware Importance Score

As mentioned in the observation section, we derive a channel calculation method called Class-Aware Importance Score (CAIS). Our importance score calculation is conducted at the class level rather than taking the whole dataset as a unity to determine the important channels. We surprisingly find some channels are commonly significant in all categories, and these channels could be used to transfer the knowledge from the pre-trained model to downstream tasks efficiently. Following previous parameter-efficient tuning algorithms (Houlsby et al., 2019; Jia et al., 2022; Hu et al., 2021; Zhang et al., 2022), we embrace VTAB-1K (Zhai et al., 2019) as our primary test dataset, which contains 1000 images for training. It is practicable to pass 1000 training images to the pre-trained model and save the intermediate feature maps for the following calculation. We define the intermediate feature maps as $\{f^l_{N_c} \in \mathbb{R}^{B_{N_c} \times L \times C} | \, l, N_c \in N, 1 \leq l \leq L \, and \, 1 \leq N_c \leq M\}$, where the L and C represent the number of tokens and amount of channel dimension, respectively. The B_{N_c} is the volume of each class and M represents the amount of categories of target dataset. The importance score vector of each layer $\{Z^l \in R^{1 \times C} | \, l \in N, \; 1 \leq l \leq 12\}$ can be formulated as:

$$Z^l = \frac{1}{M} \sum_{N_c=1}^{M} \tilde{f}^l_{N_c}, \;\; \tilde{f}^l_{N_c} \in R^{1 \times C}, \tag{1}$$

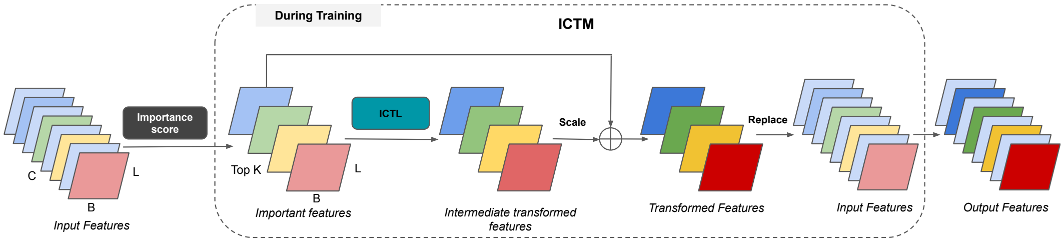


Figure 3: The overview of proposed important channel tuning module. The colorful features represent the important features, and the light blue represents the standard features.

where $\tilde{f}^l_{N_c}$ is the concatenated value of all the samples in N_c, which is described as:

$$\tilde{f}^l_{N_c} = Concat([\|f^l_{N_c,0}\|^2_2,\ \|f^l_{N_c,1}\|^2_2,\ ...,\ \|f^l_{N_c,i}\|^2_2]),\ \ f^l_{N_c,i} \in R^{B_{N_c} \times L \times 1},\ i \in N,\ 1 \leq i \leq C, \tag{2}$$

where $Concat$ means concatenate the value after $L2$ normalization operation in each channel. After getting the importance score vector Z^l, we can choose the largest K values of Z^l and then we can derive selected indices as $I^l = topK(Z^l), I^l \in N^K$ at l-th layer. Later, we will only update features of these channels with high important scores. This design enables us to tune only a subset of channels instead of all, inheriting the strong representations of pre-trained backbones for downstream tasks. For different downstream datasets, the selected important channels in each layer are different, and it could explicitly involving dataset-specific information into the model adaptation.

3.3 Adapting ViT via Important Channel Tuning Layer

Given the importance score to select top-K important channels, it is natural to tune only these channels for efficient model tuning, named as Important Channel Tuning Module (ICTM) in this paper. The overview of the ICTM is depicted in Fig. 3. Unlike other parameter-efficient tuning methods, our ICTM only contains a linear layer rather than an MLP-like adapter (Houlsby et al., 2019) or the prompt (Jia et al., 2022). This design is easy to implement, and we name this layer the Important Channel Tuning Layer (ICTL). In order to inherit the original robust representations and transform the knowledge to present task, we use a residual shortcut to add the important and intermediate transformed features and use the $Scale$ (as shown in Fig. 3) constant as a hyperparameter to determine the weights between both features. After obtaining the transformed features, we replace the important features with transformed ones and then get the final output features. Note that our ICTL is straightforward to be applied in any position at each layer. In Fig. 4, we present two forms of inserting the ICTL into the ViT. The "Ours-MLP" represents that we insert the ICTL after the MLP block, and the "Our-Attn" indicates that we put the ICTL after the MHA block while before the MLP block. In the following experiments, "Ours-Attn" is the default injection position compared with other baselines.

Discussion. We address two critical problems for adapting the large ViT models to downstream tasks, namely, avoiding overfitting and reducing the number of trainable parameters. We find that the state-of-the-art baselines lack the consideration of explicitly using the dataset-specific information. Moreover, different downstream tasks have their own peculiarities, *i.e.*, "Not all channels are equal". Thus, we propose the ICT to explicitly incorporate the dataset-specific information in the model tuning by using CAIS, which could achieve better results and simultaneously reduce the parameters. Moreover, calculating the important channels offline could save the computations of searching the best prompt length as in VPT (Jia et al., 2022) or the best network structure as in NOAH (Zhang et al., 2022).

4 Experiments

This section compares our ICT with other state-of-the-art parameter-efficient tuning baselines on the VTAB-1K benchmark, using ViT and Swin Transformer backbones. In addition, we analyze the channel selection strategy, class-aware important score, insert location, and the number of selected channels to further verify the effectiveness of ICT. Last but not least, we investigate ICT's robustness and generalization ability in domain generalization.

Table 1: Comparisons with state-of-the-art methods on the VTAB-1K benchmark with ViT-B/16. Average results are calculated across all 19 datasets. "# Params" denotes the average number of trainable parameters in the backbone. The best performance and smallest parameter number are bolded in each column.

		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**
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	65.6
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	53.0
Bias	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	62.1
VPT	0.53	**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	69.4
Adapter	0.16	69.2	90.0	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	71.4
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	72.3
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	73.3
ICT (ours)	**0.11**	75.3	91.6	**72.2**	**99.2**	**91.1**	**91.2**	**55.0**	**85.0**	**96.1**	**86.3**	**76.2**	81.5	65.1	**51.7**	80.2	75.4	46.2	**33.2**	**45.7**	**73.6**

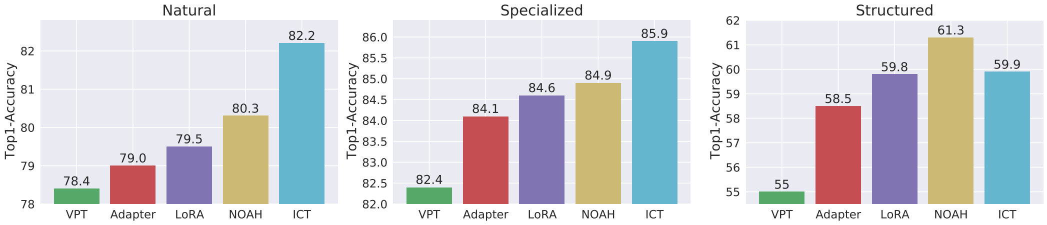


Figure 5: Group-wise average results on VTAB-1K benchmark.

4.1 Experiments on VTAB-1K Benchmark

Dataset. VTAB-1K contains 19 visual classification tasks which cover a broad spectrum of domains and semantics in three groups, *i.e.*, *Natural*, *Specialized*, and *Structured*. The *Natural* group contains 7 classic classification datasets of natural images. The *Specialized* group involves 4 datasets of two special scenarios: medical and remote-sensing. The *Structured* group has 8 datasets, mainly focusing on understanding the structure of a scene, such as object counting, and depth prediction. Each task of VTAB-1K contains 1000 training images. More details are available in the Appendix Tab. 7. Following Zhang et al. (2022), we use the 800-200 TRAIN-VAL split to determine the hyper-parameters and the entire 1000 training data to train the final model. We report the average top-1 accuracy on the TEST set.

Baselines. We compare our method with three baselines **Full fine-tuning**, **Linear**, and **Bias** without external parameters and four baselines **Adapter** (Houlsby et al., 2019), **LoRA** (Hu et al., 2021), **VPT** (Jia et al., 2022), and **NOAH** (Zhang et al., 2022) with external parameters. **Bias** method only updates all the bias terms in the pre-trained backbone. **Adapter** injects an additional MLP module into each transformer layer. **LoRA** adopts an optimized low-rank matrix to the multi-head attention module in the transformer layers. **VPT** is a visual prompt algorithm to incorporate the prompts with tokens into the backbone. **NOAH** is a neural architecture search algorithm that incorporates the Adapter, LoRA, and VPT into the network search. To provide a fair comparison, we directly borrow their released results or run their code to generate the results.

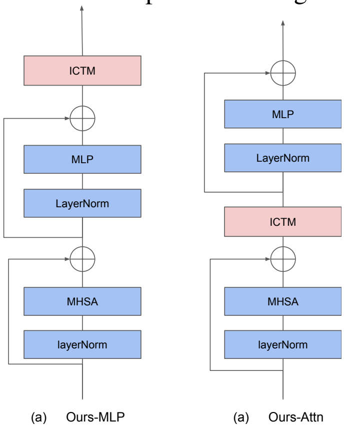


Figure 4: Two types of structures when inserting ICTM into the backbone.

Performance with ViT backbone. We compare our ICT with the above 7 baselines in Tab. 1 and Fig. 5. We use ViT-B/16 as the backbone and insert ICTM in each transformer layer. The default number of K is set to 96, 1/8 of the total channels, leading to the trainable parameter number being only 0.11M. **First**, our ICT outperforms the full fine-tuning on 18 out of 19 datasets and gains the improvement of 6.3%, 2.5%, and 12.3% in the three groups, respectively, while only additional 0.13% of the backbone parameters are learned. Such results reflect that ICT can greatly reduce the storage space and alleviate the overfitting problem commonly occurred in fine-tuning

Table 2: Comparisons with state-of-the-art methods on the VTAB-1K benchmark with pre-trained Swin-B.

Swin-B	# Params (M)	VTAB-1K			
		Natural	Specialized	Structured	Average
Full (Jia et al., 2022)	86.7	79.1	86.2	59.7	74.2
Linear (Jia et al., 2022)	0	73.5	80.8	33.5	56.4
Bias (Jia et al., 2022)	0.24	74.2	80.1	42.4	62.1
VPT (Jia et al., 2022)	0.19	76.8	84.5	53.4	68.6
ICT (ours)	**0.10**	**82.7**	**87.5**	**60.6**	**74.4**

Table 3: Evaluation of the class-aware calculation for the importance score. Backbone network is ViT-B/16. "CA" denotes Class-Aware.

ViT-B/16	# Params (M)	VTAB-1K			
		Natural	Specialized	Structured	Average
w/o CA	0.11	82.1	85.2	58.4	72.8
w/ CA	0.11	**82.2**	**85.9**	**59.9**	**73.6**

large models. **Second**, when compared with other PET methods, our ICT achieves the average accuracy of 73.6% on the 19 datasets, surpassing NOAH (Zhang et al., 2022) by 0.3% with only *a quarter of* its trainable parameters. As shown in Fig. 5, our ICT outperforms VPT (Jia et al., 2022) by 3.8%, 3.5% and 4.9% in the three groups, respectively. These results further demonstrate the effectiveness and efficiency of our ICT.

Performance with Swin Transformer Backbone. To verify the effectiveness of ICT with different backbones, we apply ICT on hierarchical transformers, *i.e.*, Swin-B. We use the same setting of inserting ICTM as in the ViT backbone: inserting the ICTM after the "Attn" block in the transformer layer. Considering deep layers contain more semantic information, instead of applying ICTM on all the transformer layers, we insert it to the last half of the layers in the stage3 and all layers of the stage4 of the Swin-B to keep a similar level of trainable parameters. The results of Tab. 2 can be found that ICT outperforms **full fine-tuning** in all three groups with only 0.13% parameters while other methods cannot. In addition, compared with PET method, ICT outperforms VPT (Jia et al., 2022) by 5.9%, 3.0%, and 7.2% in the three groups, respectively. It demonstrates that our ICT is superior to using the prompts in the Swin transformer in terms of effectiveness and efficiency. All the results suggest that our ICT is applicable for other vision transformer architectures and can achieve significant performance with only a tiny amount of trainable parameters.

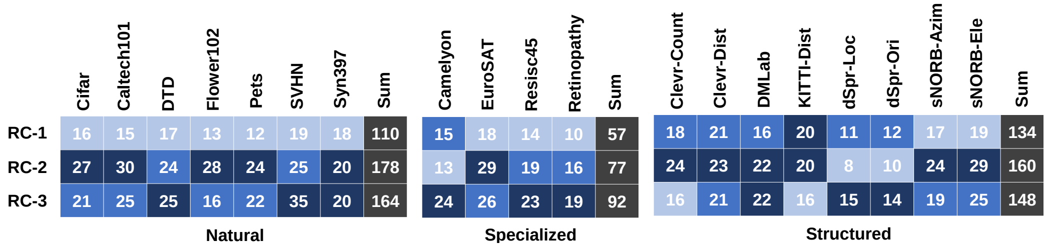
4.2 Evaluation

Effectiveness of Important Channel Tuning. We compare three channel selection strategies to verify the effectiveness of selecting important channels in Tab. 4. The selection strategies include Important Channel Selection (IC), Unimportant Channel Selection (UC), and Random Channel Selection (RC). We randomly select three sets of channels (RC-1/2/3) for random channel selection to alleviate the outliers. As shown in Tab. 4, IC achieves the best results and outperforms UC by 2.4% in the average accuracy. Random channel selection could obtain modest results compared with full fine-tuning, but all of them perform worse than IC. Interestingly, UC, RC, and IC can perform better than full fine-tuning, demonstrating that selecting a small subset of the channels can prevent the large model from overfitting to the small training set. Full results on 19 datasets are placed in Tab. 9. To reflect the effect of selecting important channels, we calculate the overlaps of indices between the important channels and randomly selected channels depicted in Tab. 4. As the overlaps between RC and IC are small, it is valid to explain the performance drop of RC compared to IC. Nevertheless, some results suggest that higher overlaps of RC-2 exhibit better performance than RC-1 and RC-3 in *Natural* and *Structured* groups.

Effectiveness of Class-Aware Calculation for importance score. As mentioned in Sec. 3.2, our method considers the effect of each class rather than taking the dataset as a whole to estimate the importance score (IS). This design aims to find the important channels in all categories rather than the entire training set. When meeting a new downstream task, the user can sample the images in

Table 4: Comparison of different channel selection strategies.

ViT-B/16	# Params (M)	VTAB-1K			
		Natural	Specialized	Structured	Average
Full	85.8	75.9	83.4	47.6	65.6
RC-1	0.11	81.2	85.3	56.5	71.7
RC-2		81.9	85.0	56.8	72.0
RC-3		81.0	85.1	56.2	71.4
UC		80.7	85.7	55.7	71.2
IC		**82.2**	**85.9**	**59.9**	**73.6**



	Cifar	Caltech101	DTD	Flower102	Pets	SVHN	Syn397	Sum
RC-1	16	15	17	13	12	19	18	110
RC-2	27	30	24	28	24	25	20	178
RC-3	21	25	25	16	22	35	20	164

Natural

	Camelyon	EuroSAT	Resisc45	Retinopathy	Sum
RC-1	15	18	14	10	57
RC-2	13	29	19	16	77
RC-3	24	26	23	19	92

Specialized

	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	Sum
RC-1	18	21	16	20	11	12	17	19	134
RC-2	24	23	22	20	8	10	24	29	160
RC-3	16	21	22	16	15	14	19	25	148

Structured

Figure 6: Correlation analysis between the performance and channel selection in Random Channel selection strategy (RC). RC-1, RC-2, and RC-3 represent three random selection experiments with different random seeds. We compare the randomly generated channels with the important ones calculated by importance score and record the sum of the overlaps in each transformer layer per task. Darker color means a higher value.

each class to calculate CAIS to reduce the storage of using the entire training set. As shown in Tab. 3, adopting class-aware calculation can yield improvement in all three groups.

Insert Position. As shown in Fig. 4, ICTM can be inserted after MLP block (ICT_{MLP}) or between MHA block and MLP block (ICT_{Attn}). To investigate the influence of insert location, we compare two forms on the VTAB-1K benchmark in Tab. 5. Both achieve promising performances, and ICT_{Attn} outperforms ICT_{MLP} on two of the three groups. The main reason is probably that after gathering long-range dependencies with MHA, the features contain more salient and important channels, which can be better adapted to downstream tasks.

Number of selected channels K. The most important hyperparameter of ICTM is the number of selected channels K, which influences the model architecture and the number of trainable parameters. **Note that** different from previous works (Jia et al., 2022; Zhang et al., 2022) that select hyperparameters for each dataset, we use the same K for all the datasets. As shown in Tab. 12 in the appendix, $\text{ICT}_{\text{K=32}}$ beats the full fine-tuning and bias tuning with the improvements of 3.5% and 7.0%, respectively. Specifically, compared with tuning bias, $\text{ICT}_{\text{K=32}}$ only adopts 0.01M parameters as tuning bias term adopts 0.10M. As shown in Fig. 7, the performance generally improves along with the increase of K. However, the improvement of $K = 192$ over $K = 96$ is marginal, while the number of parameters is four times larger. Considering both the effectiveness and efficiency, we set K to 96 by default.

4.3 Experiments on Domain Generalization

Dataset. In addition to evaluating the model on test data of the same distribution, modern deep neural networks commonly suffer from performance degradation when the testing distribution is

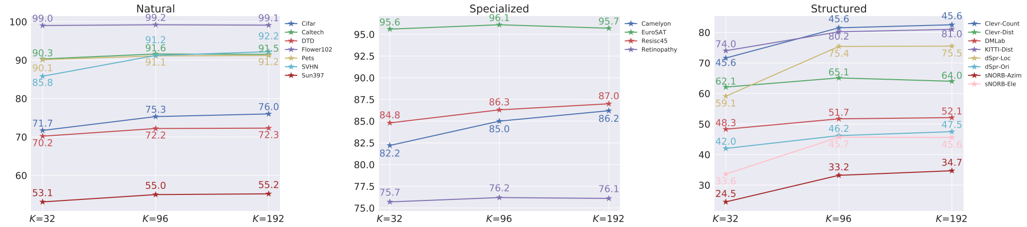


Figure 7: Evaluation of choosing different K values. Zoom in for better view.

Table 5: Comparison of different insert positions. Backbone network is ViT-B/16.

ViT-B/16	# Params (M)	VTAB-1K			
		Natural	Specialized	Structured	Average
ICT_{MLP}	0.11	81.7	**86.0**	58.6	72.9
ICT_{Attn}	0.11	**82.2**	85.9	**59.9**	**73.6**

Table 6: Comparison with previous methods on domain generalization. Backbone network is ViT-B/16. The number of important channels is 192.

	Source	**Target**			
	ImageNet	-V2	-Sketch	-A	-R
Adapter (Houlsby et al., 2019)	70.5	59.1	16.4	5.5	22.1
VPT (Jia et al., 2022)	70.5	58.0	18.3	4.6	23.2
LoRA (Hu et al., 2021)	70.8	59.3	20.0	6.9	23.3
NOAH (Zhang et al., 2022)	71.5	**66.1**	24.8	11.9	28.5
ICT-B (ours)	**77.1**	65.8	**28.5**	**12.1**	**31.0**

different from that of the training set, *i.e.*, domain shift, which is inevitable in a real-world application. To alleviate this problem, domain generalization (Zhou et al., 2021b; Zhao et al., 2022) is investigated in the community, which aims at training a model with one or multiple source domains but can perform well on other unseen target domains. To verify the generalization ability of our ICT, we follow Zhang et al. (2022) to conduct experiments on ImageNet and its variants. Specifically, we use the ImageNet-1K (Deng et al., 2009) as the source domain with 16-shot per category and evaluate our model on ImageNetV2 (Recht et al., 2019), ImageNet-Sketch (Wang et al., 2019), ImageNet-A (Hendrycks et al., 2021b), and ImageNet-R (Hendrycks et al., 2021a). ImageNetV2 (Recht et al., 2019) is collected from different sources from ImageNet-1K with the same protocol, and ImageNet-Sketch (Wang et al., 2019) contains the sketch images of ImageNet classes. Both of them use the same classes as ImageNet-1K. ImageNet-A (Hendrycks et al., 2021b) and ImageNet-R (Hendrycks et al., 2021a) contains the adversarially-filtered images and renditions of ImageNet data of a 200-class subset, respectively. We use a large version of ICT, *i.e.*, ICT-B, containing a comparable number of parameters with NOAH (0.44M *vs.*0.43M).

Results. In Tab. 6, we compare our ICT-B with Adapter (Houlsby et al., 2019), VPT (Jia et al., 2022), LoRA (Hu et al., 2021), and NOAH (Zhang et al., 2022) on the above datasets to verify the generalization ability. We can make two observations. **First**, ICT-B outperforms the previous best method (NOAH) on three of the four target datasets and achieves comparable performance on ImageNetV2. Specifically, ICT-B yields an improvement of 2.5% on ImageNet-R over NOAH. **Second**, our ICT-B achieves an accuracy of 77.1% on the source domain, greatly outperforming previous methods by 6%. Since the backbone model is pre-trained on ImageNet-21K, the results on ImageNet-1K show that ICT can better enhance the knowledge transfer from superset to subset. The two observations demonstrate the superiority of our ICT over previous fine-tuning techniques on strong generalization ability.

5 Conclusion

In this paper, we propose a novel parameter-efficient tuning algorithm, dubbed Important Channel Tuning (ICT), to effectively adapt the knowledge from large pre-trained ViT models to the downstream tasks. Considering the variety of channels in the extracted features from a backbone model, we design a criteria of importance score to determine the most informative channels for effective adaptation, which significantly alleviates the overfitting problem and saves storage space. Equipped with ICT, the subset parameter fine-tuned model surpasses full fine-tuning on most datasets of VTAB-1K benchmark by adding tiny number of parameters. Moreover, compared with previous PET methods, ICT achieves the state-of-the-art average performance on the VTAB-1K benchmark with both ViT and Swin Transformer backbones.

REFERENCES

Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. In *NeurIPS*, 2021. 2

Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016. 14

Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. Tinytl: Reduce memory, not parameters for efficient on-device learning. In *NeurIPS*, 2020. 2

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020. 1

Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 357–366, 2021a. 2

Hao Chen, Ran Tao, Han Zhang, Yidong Wang, Wei Ye, Jindong Wang, Guosheng Hu, and Marios Savvides. Conv-adapter: Exploring parameter efficient transfer learning for convnets. *arXiv preprint arXiv:2208.07463*, 2022a. 3

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *arXiv preprint arXiv:2205.13535*, 2022b. 2, 3

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9640–9649, 2021b. 2

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 14

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 14

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. Ieee, 2009. 9, 14

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12124–12134, 2022. 2

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. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pp. 2286–2296. PMLR, 2021. 2

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6824–6835, 2021. 2

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *conference on computer vision and pattern recognition workshop*, pp. 178–178. IEEE, 2004. 3, 14

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 14

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. In *NeurIPS*, 2021. 2

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *NeurIPS*, 2015. 3

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. 14

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8349, 2021a. 9, 14

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15262–15271, 2021b. 9, 14

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 (ICML)*, pp. 2790–2799. PMLR, 2019. 2, 3, 4, 5, 6, 9

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3, 4, 6, 9

Menglin Jia, Zuxuan Wu, Austin Reiter, Claire Cardie, Serge Belongie, and Ser-Nam Lim. Exploring visual engagement signals for representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4206–4217, 2021. 2

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022. 2, 3, 4, 5, 6, 7, 8, 9, 14

Shibo Jie and Zhi-Hong Deng. Convolutional bypasses are better vision transformer adapters. *arXiv preprint arXiv:2207.07039*, 2022. 2, 3

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2901–2910, 2017. 14

Kaggle and EyePacs. Kaggle diabetic retinopathy detection. 2015. URL https://www.kaggle.com/c/diabetic-retinopathy-detection/data. 14

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 14

Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pp. II–104. IEEE, 2004. 14

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rJqFGTslg. 3

Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*, 2021. 1

Lingbo Liu, Bruce XB Yu, Jianlong Chang, Qi Tian, and Chang-Wen Chen. Prompt-matched semantic segmentation. *arXiv preprint arXiv:2208.10159*, 2022. 3

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021a. 2

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021b. 1, 2

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *International Conference on Learning Representations*, 2018. 3

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 181–196, 2018. 2

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset, 2017. 14

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 14

Xing Nie, Bolin Ni, Jianlong Chang, Gaomeng Meng, Chunlei Huo, Zhaoxiang Zhang, Shiming Xiang, Qi Tian, and Chunhong Pan. Pro-tuning: Unified prompt tuning for vision tasks. *arXiv preprint arXiv:2207.14381*, 2022. 3

M-E Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pp. 1447–1454. IEEE, 2006. 14

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3498–3505. IEEE, 2012. 14

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 13

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021. 2

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, pp. 5389–5400. PMLR, 2019. 9, 14

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL https://openreview.net/forum?id=Zkj_VcZ6ol. 1

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7262–7272, 2021. 1

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017. 1

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 32–42, 2021. 2

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 210–218. Springer, 2018. 14

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019. 9, 14

Pichao Wang, Xue Wang, Fan Wang, Ming Lin, Shuning Chang, Wen Xie, Hao Li, and Rong Jin. Kvt: k-nn attention for boosting vision transformers. *arXiv preprint arXiv:2106.00515*, 2021. 2

Shijie Wang, Jianlong Chang, Zhihui Wang, Haojie Li, Wanli Ouyang, and Qi Tian. Fine-grained retrieval prompt tuning. *arXiv preprint arXiv:2207.14465*, 2022. 3

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010. 14

Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, and Yanning Zhang. Class-aware visual prompt tuning for vision-language pre-trained model. *arXiv preprint arXiv:2208.08340*, 2022. 3

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 558–567, 2021. 2

Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

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. 2, 4, 14

Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. Neural prompt search. *arXiv preprint arXiv:2206.04673*, 2022. 3, 4, 5, 6, 7, 8, 9, 14

Yuyang Zhao, Zhun Zhong, Na Zhao, Nicu Sebe, and Gim Hee Lee. Style-hallucinated dual consistency learning for domain generalized semantic segmentation. In *ECCV*, 2022. 9

Zangwei Zheng, Xiangyu Yue, Kai Wang, and Yang You. Prompt vision transformer for domain generalization. *arXiv preprint arXiv:2208.08914*, 2022. 3

Jingkai Zhou, Pichao Wang, Fan Wang, Qiong Liu, Hao Li, and Rong Jin. Elsa: Enhanced local self-attention for vision transformer. *arXiv preprint arXiv:2112.12786*, 2021a. 1, 2

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021b. 9

A Appendix

We use PyTorch (Paszke et al., 2017) to implement all experiments on NVIDIA V100-32GB GPUs.

A.1 Datasets Detail

Tab. 7 summarize the details of the evaluated classification datasets.

Table 7: Specifications of used datasets.

	Dataset	# Classes	Train	Val	Test
	VTAB-1K (Zhai et al., 2019)				
Natural	CIFAR100 (Krizhevsky et al., 2009)	100	800/1000	200	10,000
	Caltech101 (Fei-Fei et al., 2004)	102			6,048
	DTD (Cimpoi et al., 2014)	47			1,880
	Oxford-Flowers102 (Nilsback & Zisserman, 2006)	192			6,149
	Oxford-PetS (Parkhi et al., 2012)	37			3,669
	SVHN (Netzer et al., 2011)	10			26,032
	Sun397 (Xiao et al., 2010)	397			21,750
Specialized	Patch Camelyon (Veeling et al., 2018)	2	800/1000	200	32,768
	EuroSAT (Helber et al., 2019)	10			5,400
	Resisc45 (Cheng et al., 2017)	45			1,880
	Retinopathy (Kaggle & EyePacs, 2015)	5			42,670
Structured	Clevr/count (Johnson et al., 2017)	8	800/1000	200	15,000
	Clevr/distance (Johnson et al., 2017)	6			15,000
	DMLab (Beattie et al., 2016)	6			22,735
	KITTI-Dist (Geiger et al., 2013)	4			711
	dSprites/location (Matthey et al., 2017)	16			73,728
	dSprites/orientation (Matthey et al., 2017)	16			73,728
	SmallNORB/azimuth (LeCun et al., 2004)	18			12,150
	SmallNORB/elevation (LeCun et al., 2004)	18			12,150
	Domain generalization (Zhai et al., 2019)				
	ImageNet-1K (Deng et al., 2009)	1,000	16 per class	50,000	N/A
	ImageNet-V2 (Recht et al., 2019)	1,000	N/A	N/A	10,000
	ImageNet-Sketch (Wang et al., 2019)	1,000	N/A	N/A	50,889
	ImageNet-A (Hendrycks et al., 2021b)	200	N/A	N/A	7,500
	ImageNet-R (Hendrycks et al., 2021a)	200	N/A	N/A	30,000

A.2 Supplementary Results

Tab. 8 shows the per-task results on VTAB-1K benchmark evaluated in Tab. 2. We noticed that only the average results of three groups in VTAB-1K are presented in VPT (Jia et al., 2022); hence, we present our full results for other researchers to compare.

Tab. 9 presents the per-task results on VTAB-1K benchmark for the detail comparison. As shown in the table, the results of RC-1, RC-2, and RC-3 are unstable, so selecting the channels in a feature map affects the final accuracy. Therefore, choosing channels properly is important to achieve promising performance.

Tab. 10 shows the per-task results on the VTAB-1K benchmark of evaluating the effectiveness by adopting the class-aware importance score in the calculation of channel selection. In most tasks, w/ CA acquires higher results than w/o CA.

Tab. 11 and Tab. 12 present the per-task results on VTAB-1K for the ablation study.

More plot curves of importance score on each task of VTAB-1K can be found from Fig. 27 to Fig. 45.

A.3 Evaluating the changes of selecting important channels.

As we select the important channels before downstream training to avoid the computations to search the prompts or best architecture as in VPT (Jia et al., 2022) and NOAH (Zhang et al., 2022), we use the extracted features from the fixed backbone to guide the selection of important channels. One possible question is: Will the important channels be changed adaptively during the training? To answer this question, we load the trained models to calculate the important channels again per task. The comparison between the before and after training is shown from Fig. 8 to Fig. 26. In most tasks, the curves of before and after training are overlapped in all 12 transformer layers, but some tasks *i.e.*, *Diabetic Retinopathy*, *KITTI*, *dSprites-orientation*, *SmallNORB-azimuth* and *SmallNORB-elevation* show the small difference at deep layers. These tasks are not medical image classification, depth prediction, and orientation prediction problems that deviate from the pre-trained task heavily.

Table 8: Per-task results on VTAB-1K benchmark of Tab. 2 with a pre-trained Swin-B. Average result is calculated on all 19 datasets in VTAB-1K benchmark.

		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**
ICT	0.10	75.7	92.6	76.5	99.7	91.7	87.2	55.5	87.6	96.5	89.4	76.6	82.5	63.1	53.7	85.9	86.7	46.1	26.8	40.0	74.4

Table 9: Per-task results on VTAB-1K benchmark of Tab. 4 with a pre-trained ViT-B/16. Average result is calculated on all 19 datasets in VTAB-1K benchmark.

		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**
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	65.6
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	53.0
UC	0.11	68.8	91.5	71.5	99.0	91.1	89.0	54.3	85.3	95.8	86.6	75.2	78.1	61.2	50.3	79.0	73.1	40.0	27.4	36.7	71.1
RC-1	0.11	72.1	90.4	71.7	99.1	91.2	90.2	54.0	84.2	95.5	85.7	75.9	78.8	62.5	49.5	77.6	72.2	43.3	27.3	40.9	71.7
RC-2	0.11	74.1	91.4	72.1	99.2	90.9	90.2	55.1	83.0	95.7	86.3	75.0	79.3	62.5	50.4	78.2	72.5	43.2	28.1	39.8	72.0
RC-3	0.11	71.3	91.1	70.9	99.0	90.9	89.6	54.4	83.9	95.7	85.9	74.8	78.6	62.2	48.9	80.2	71.2	42.1	27.8	38.5	71.4
ICT	0.11	75.3	91.6	72.2	99.2	91.1	91.2	55.0	85.0	96.1	86.3	76.2	81.5	65.1	51.7	80.2	75.4	46.2	33.2	45.7	73.6

Table 10: Per-task results of evaluating the effectiveness of adopting the Class-Aware Importance Score as in Tab. 3. Average result is calculated on all 19 datasets in VTAB-1K benchmark.

		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**
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	65.6
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	53.0
w/o CA	0.11	75.1	91.8	71.6	99.1	90.9	91.2	55.1	84.4	95.6	85.6	75.1	81.1	64.7	51.2	78.3	75.0	44.8	31.3	40.4	72.8
w/ CA	0.11	75.3	91.6	72.2	99.2	91.1	91.2	55.0	85.0	96.1	86.3	76.2	81.5	65.1	51.7	80.2	75.4	46.2	33.2	45.7	73.6

Table 11: Per-task results of evaluating the effectiveness of the insert positions as in Tab. 5. Average result is calculated on all 19 datasets in VTAB-1K benchmark.

		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**
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	65.6
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	53.0
ICT$_{\text{MLP}}$	0.11	72.9	92.0	71.6	99.1	90.6	90.5	54.9	86.2	95.9	85.9	76.0	81.3	64.7	50.7	78.9	77.2	43.3	29.3	43.6	72.9
ICT$_{\text{Attn}}$	0.11	75.3	91.6	72.2	99.2	91.1	91.2	55.0	85.0	96.1	86.3	76.2	81.5	65.1	51.7	80.2	75.4	46.2	33.2	45.7	73.6

Table 12: Per-task results of evaluating different K values on VTAB-1K benchmark.

		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**
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	65.6
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	53.0
Bias	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	62.1
ICT$_{\text{K=32}}$	0.01	71.7	90.3	70.2	99.0	90.1	85.8	53.1	82.2	95.6	84.8	75.7	71.6	62.1	48.3	74.0	59.1	42.0	24.5	33.6	69.1
ICT$_{\text{K=96}}$	0.11	75.3	91.6	72.2	99.2	91.1	91.2	55.0	85.0	96.1	86.3	76.2	81.5	65.1	51.7	80.2	75.4	46.2	33.2	45.7	73.6
ICT$_{\text{K=192}}$	0.44	76.0	91.5	72.3	99.1	91.2	92.2	55.2	86.2	95.7	87.0	76.1	82.5	64.0	52.1	81.0	75.5	47.5	34.7	45.6	74.0

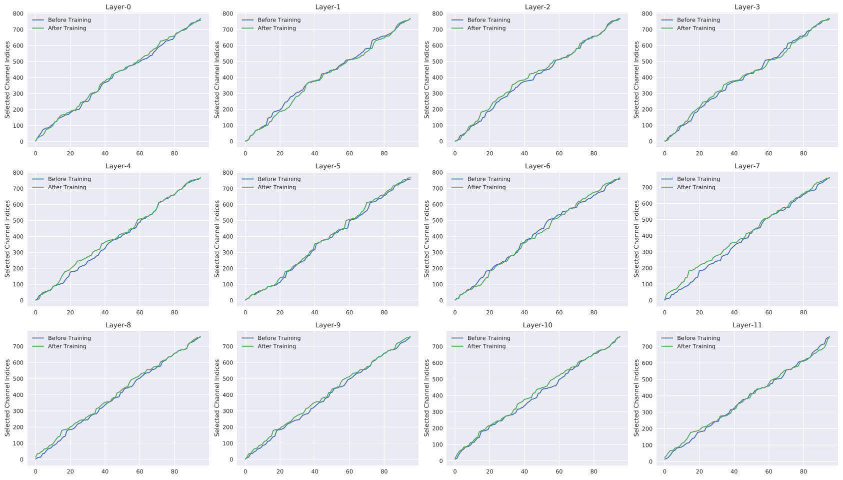


Figure 8: The difference of the calculated important channels between before and after training on CIFAR100 with a pre-trained ViT-B/16.

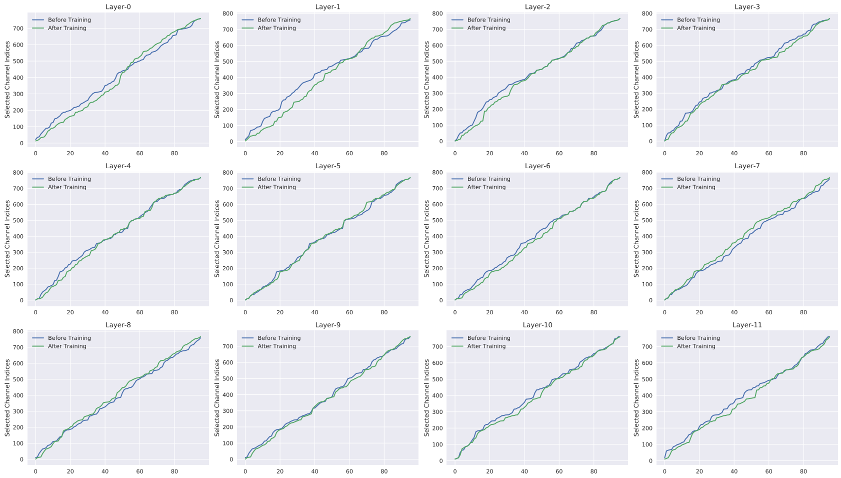


Figure 9: The difference of the calculated important channels between before and after training on Caltech101 with a pre-trained ViT-B/16.

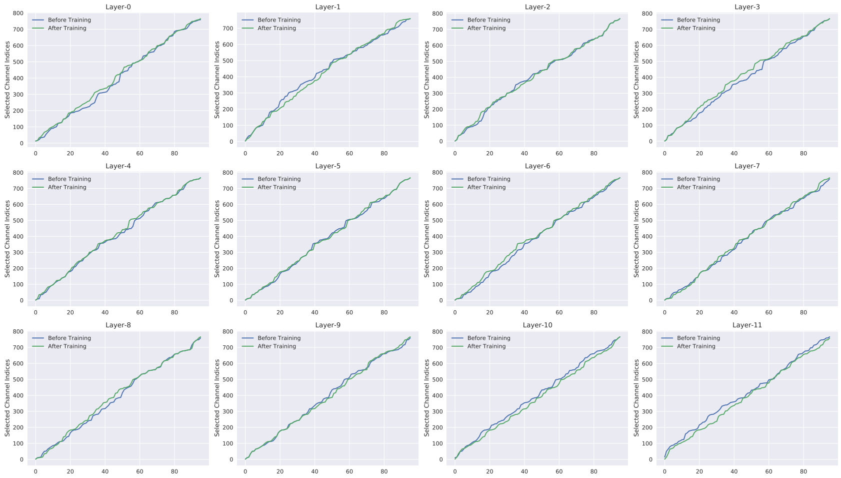


Figure 10: The difference of the calculated important channels between before and after training on DTD with a pre-trained ViT-B/16.

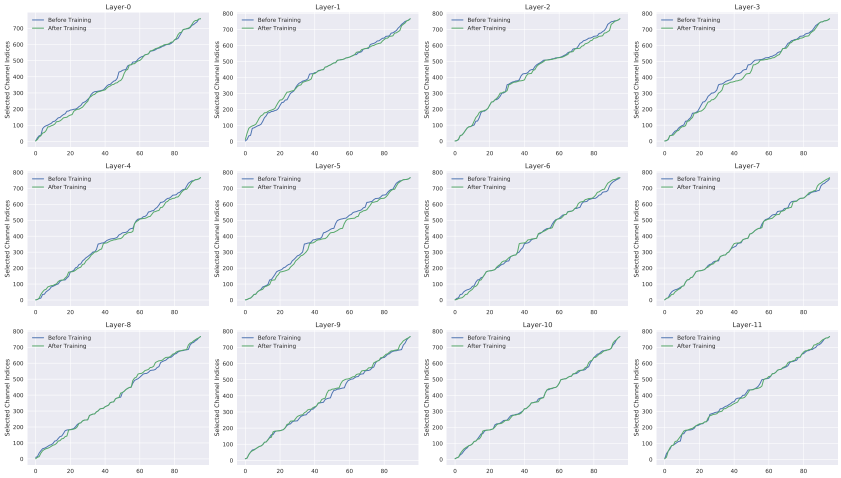


Figure 11: The difference of the calculated important channels between before and after training on Oxford-Flowers102 with a pre-trained ViT-B/16.

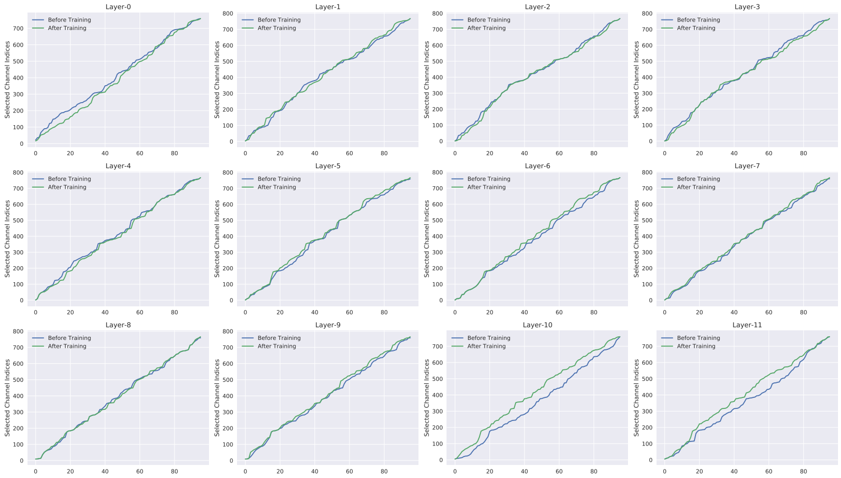


Figure 12: The difference of the calculated important channels between before and after training on Oxford-Pets with a pre-trained ViT-B/16.

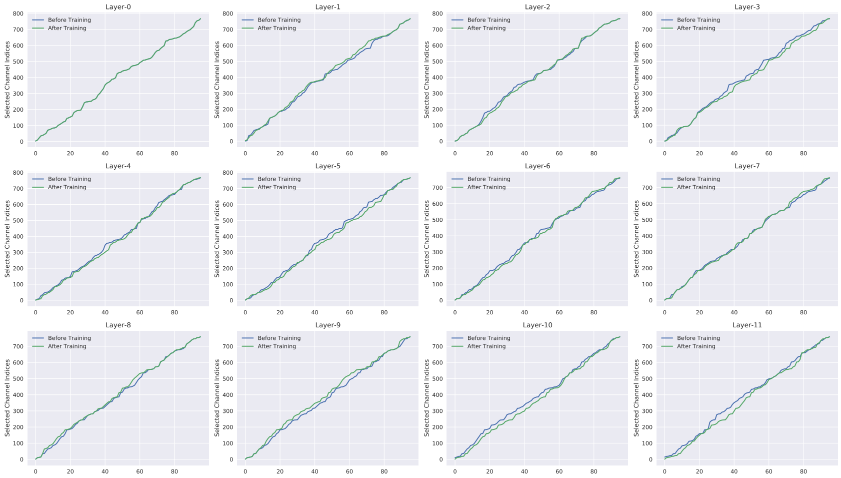


Figure 13: The difference of the calculated important channels between before and after training on SVHN with a pre-trained ViT-B/16.

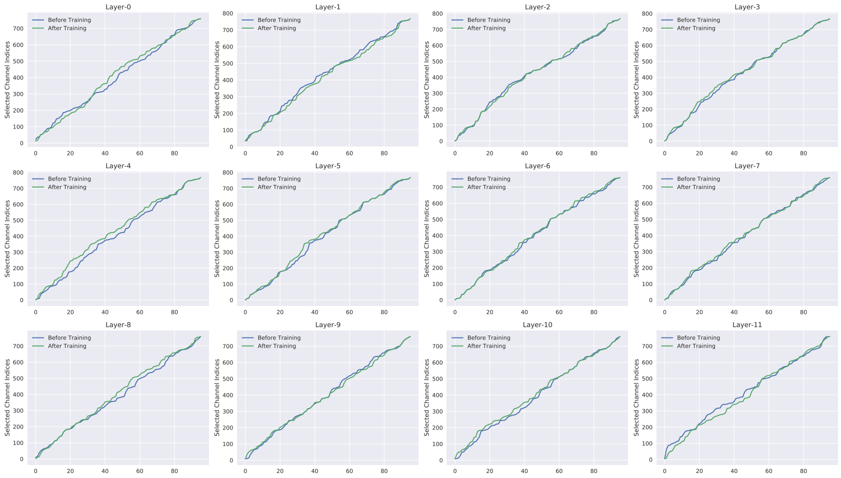


Figure 14: The difference of the calculated important channels between before and after training on Sun397 with a pre-trained ViT-B/16.

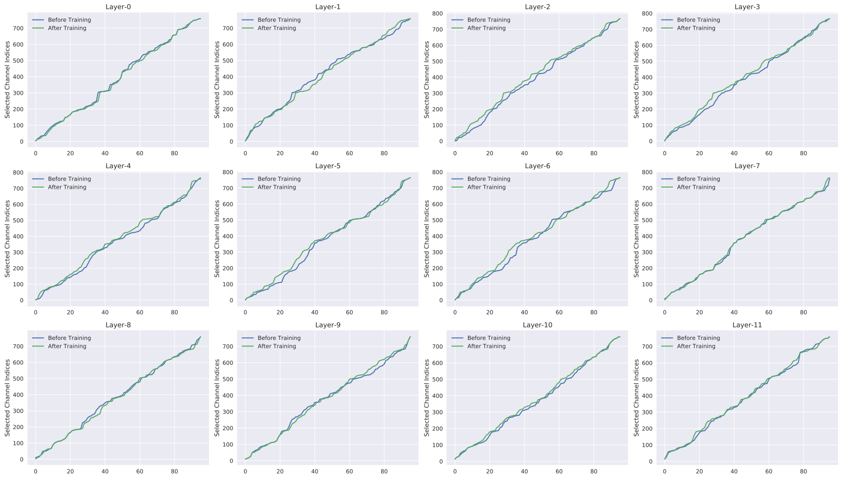


Figure 15: The difference of the calculated important channels between before and after training on Patch Camelyon with a pre-trained ViT-B/16.

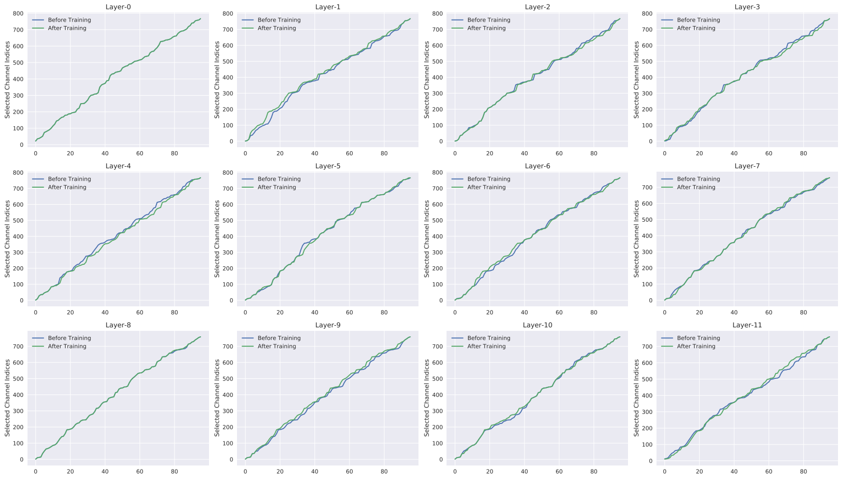


Figure 16: The difference of the calculated important channels between before and after training on EuroSAT with a pre-trained ViT-B/16.

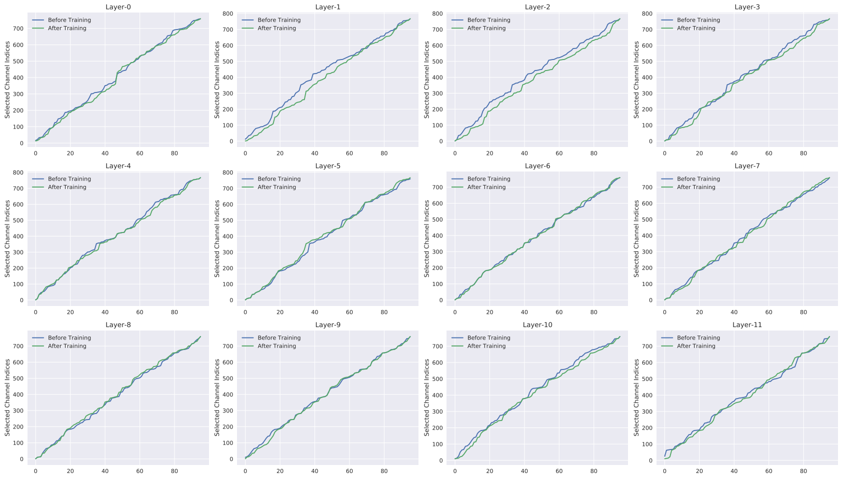


Figure 17: The difference of the calculated important channels between before and after training on Resisc45 with a pre-trained ViT-B/16.

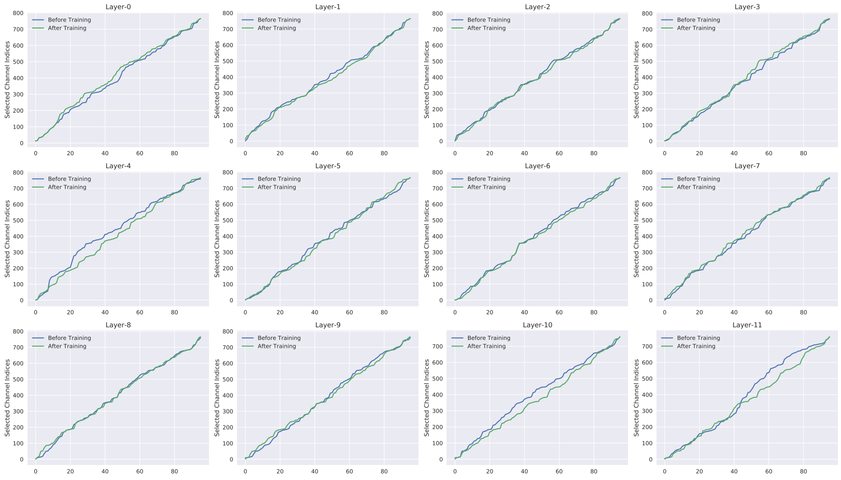


Figure 18: The difference of the calculated important channels between before and after training on Diabetic Retinopathy with a pre-trained ViT-B/16.

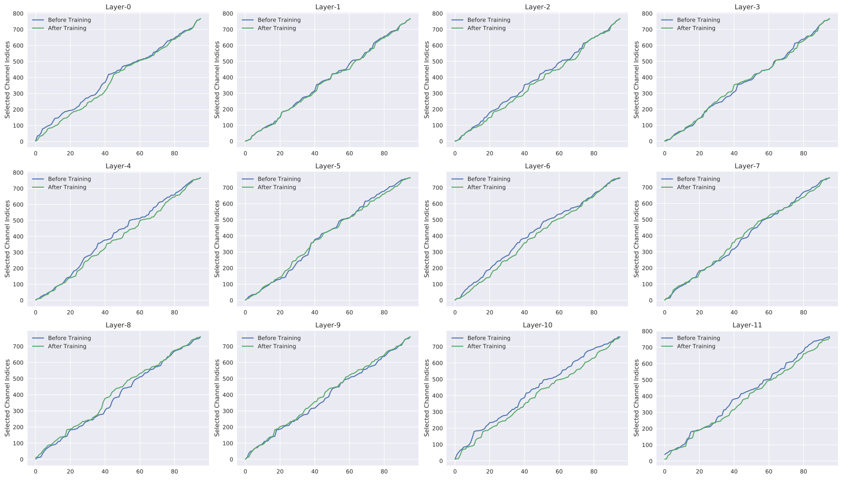


Figure 19: The difference of the calculated important channels between before and after training on Clevr-count with a pre-trained ViT-B/16.

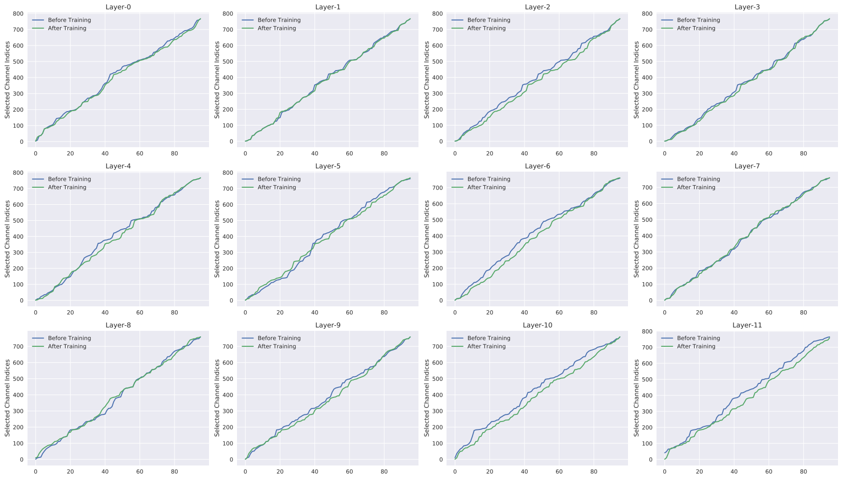


Figure 20: The difference of the calculated important channels between before and after training on Clevr-distance with a pre-trained ViT-B/16.

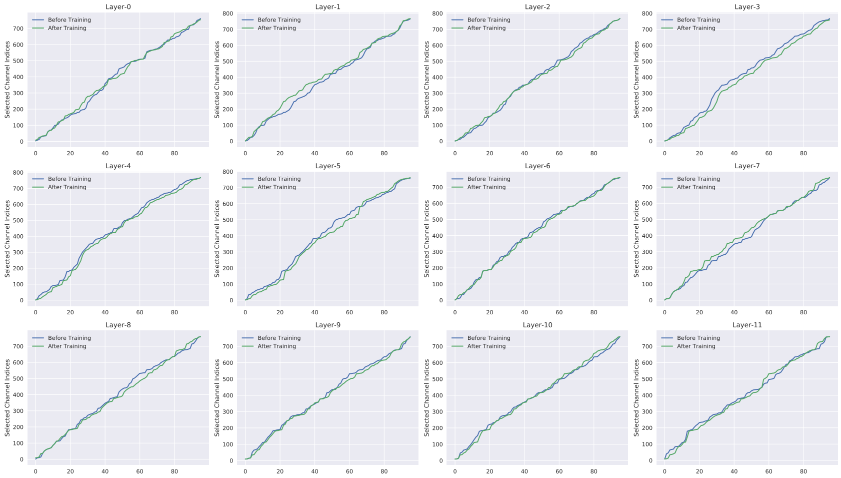


Figure 21: The difference of the calculated important channels between before and after training on DMLab with a pre-trained ViT-B/16.

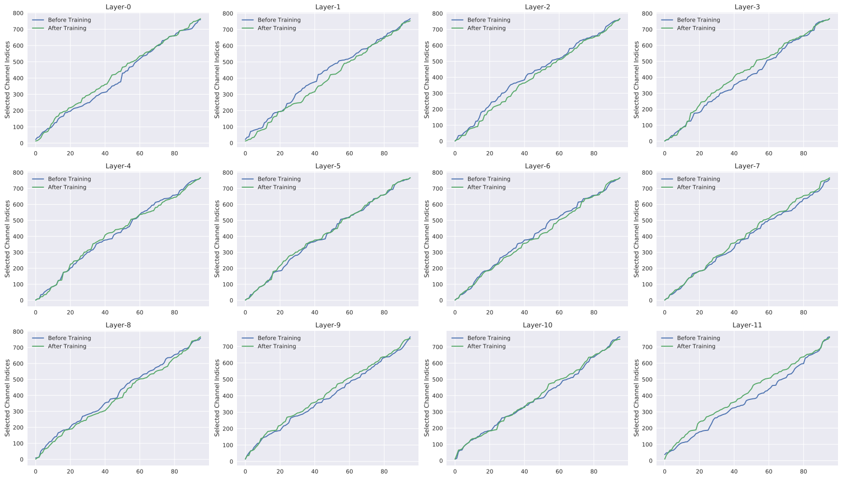


Figure 22: The difference of the calculated important channels between before and after training on KITTI with a pre-trained ViT-B/16.

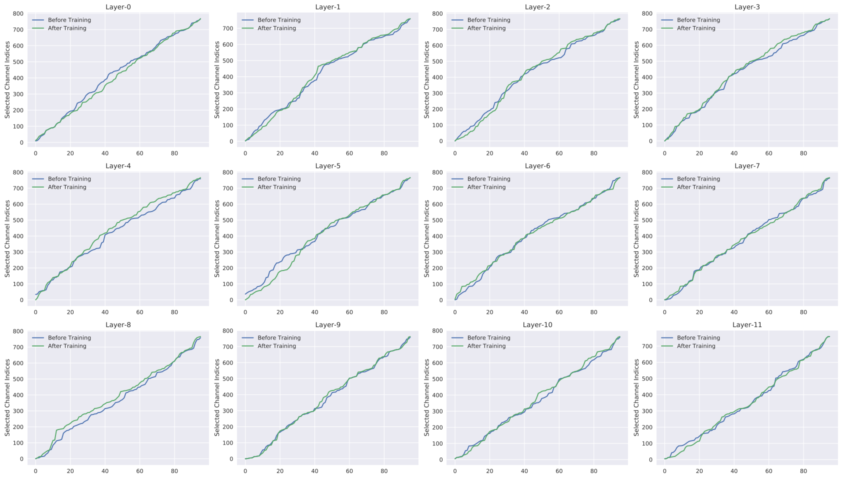


Figure 23: The difference of the calculated important channels between before and after training on dSprites-location with a pre-trained ViT-B/16.

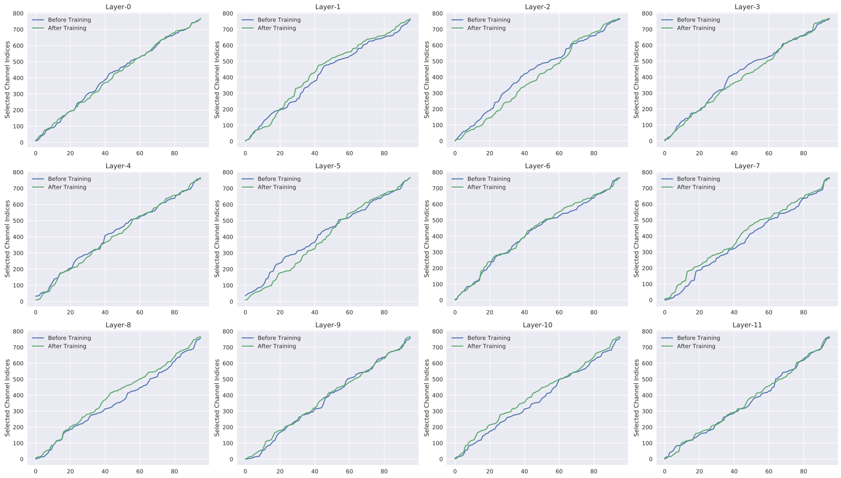


Figure 24: The difference of the calculated important channels between before and after training on dSprites-orientation with a pre-trained ViT-B/16.

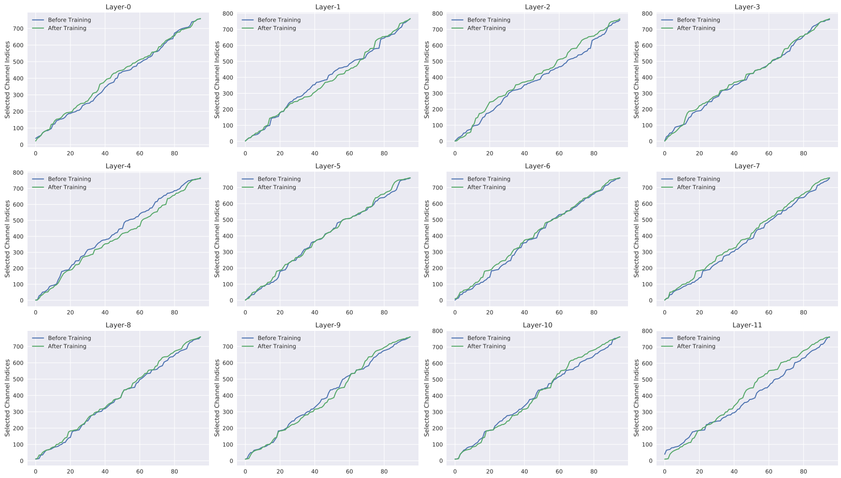


Figure 25: The difference of the calculated important channels between before and after training on SmallNORB-azimuth with a pre-trained ViT-B/16.

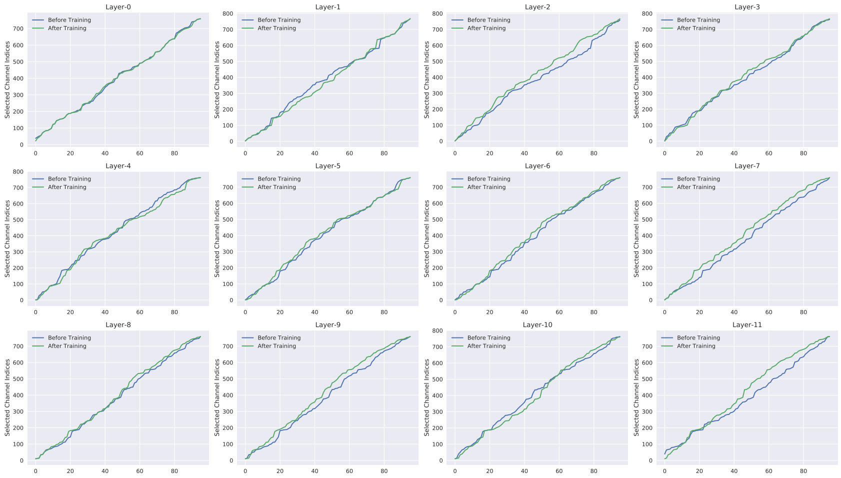


Figure 26: The difference of the calculated important channels between before and after training on SmallNORB-elevation with a pre-trained ViT-B/16.

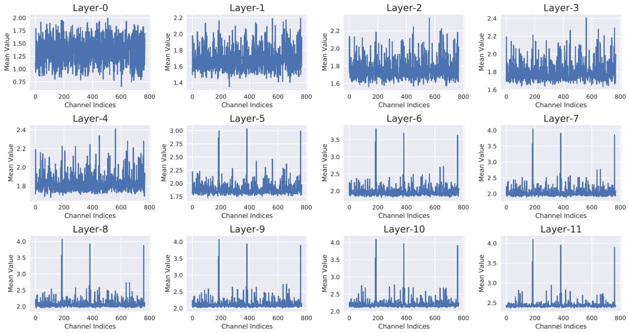


Figure 27: The plot curves of the important scores calculated on CIFAR100 at each transformer layer.

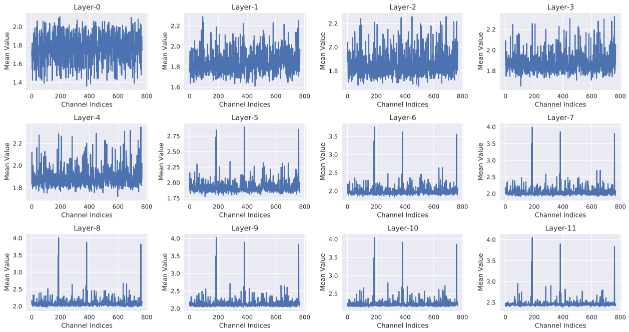


Figure 28: The plot curves of the important scores calculated on Caltech101 at each transformer layer.

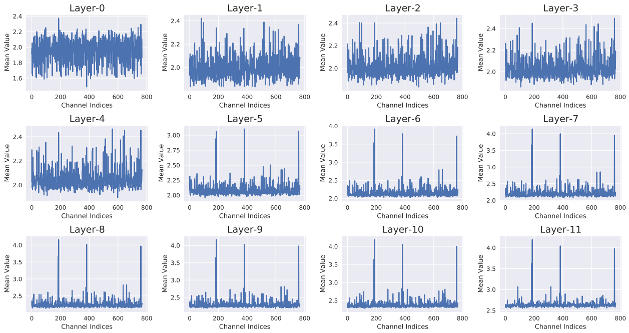


Figure 29: The plot curves of the important scores calculated on DTD at each transformer layer.

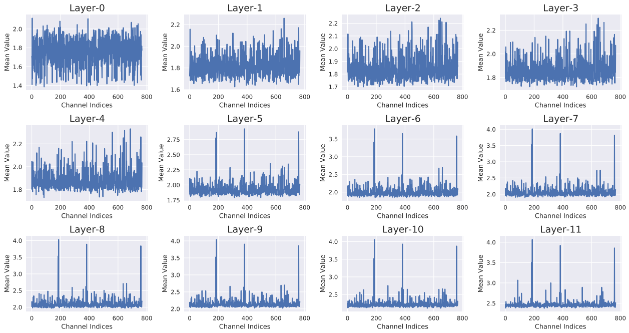


Figure 30: The plot curves of the important scores calculated on Oxford-Flowers102 at each transformer layer.

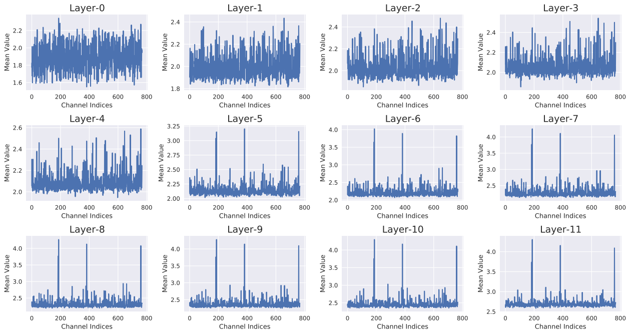


Figure 31: The plot curves of the important scores calculated on Oxford-Pets at each transformer layer.

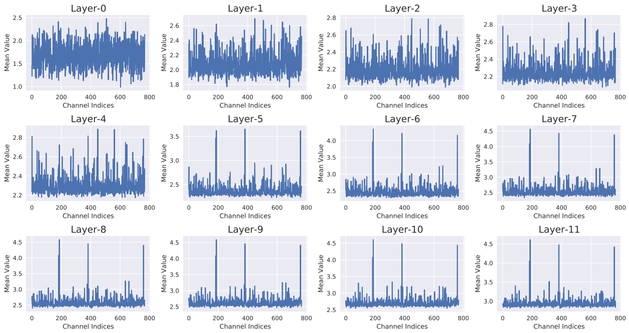


Figure 32: The plot curves of the important scores calculated on SVHN at each transformer layer.

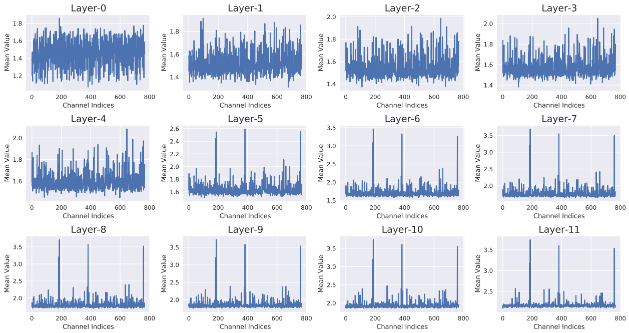


Figure 33: The plot curves of the important scores calculated on Sun397 at each transformer layer.

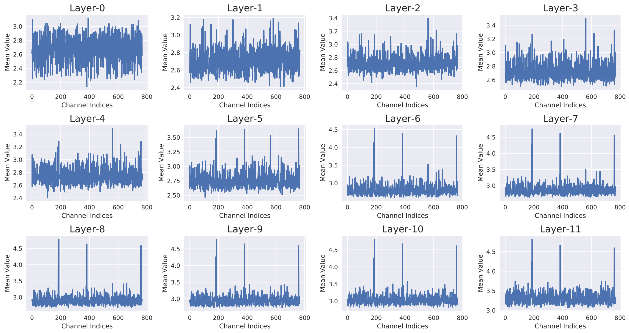


Figure 34: The plot curves of the important scores calculated on Patch Camelyon at each transformer layer.

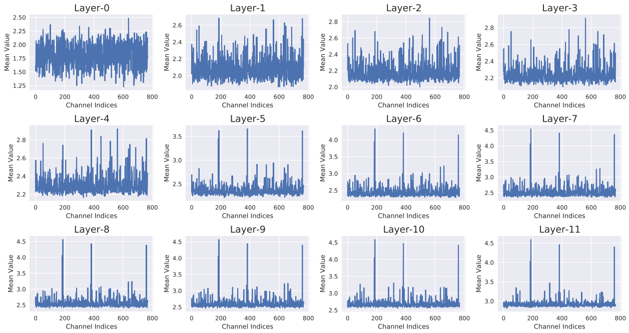


Figure 35: The plot curves of the important scores calculated on EuroSAT at each transformer layer.

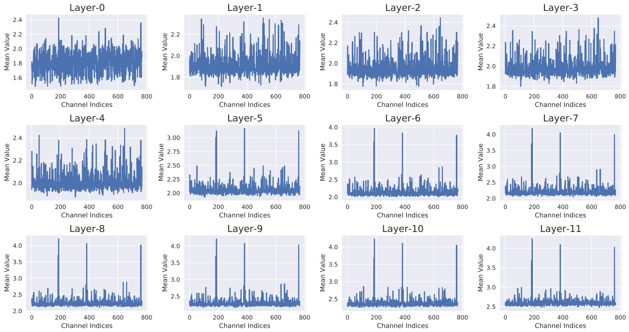


Figure 36: The plot curves of the important scores calculated on Resisc45 at each transformer layer.

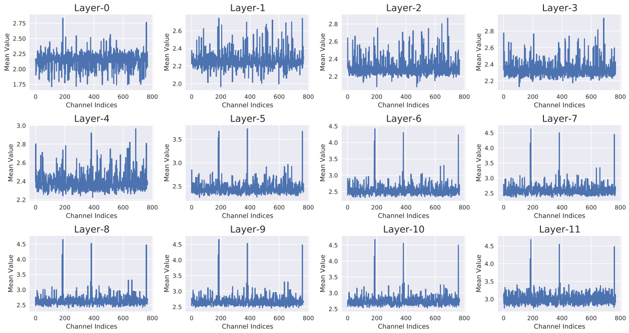


Figure 37: The plot curves of the important scores calculated on Diabetic Retinopathy at each transformer layer.

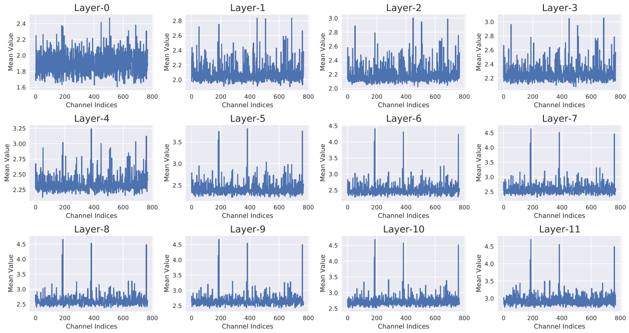


Figure 38: The plot curves of the important scores calculated on Clevr-count at each transformer layer.

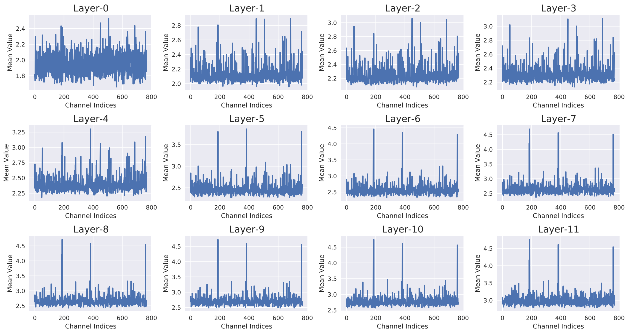


Figure 39: The plot curves of the important scores calculated on Clevr-distance at each transformer layer.

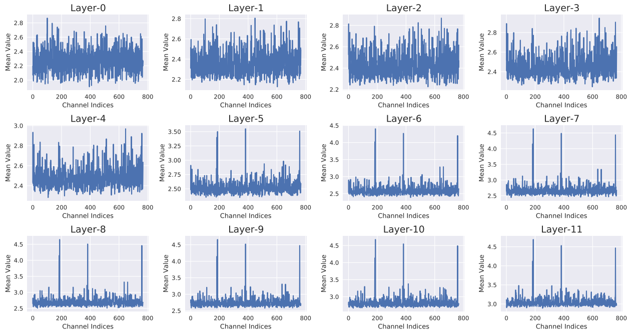


Figure 40: The plot curves of the important scores calculated on DMLab at each transformer layer.

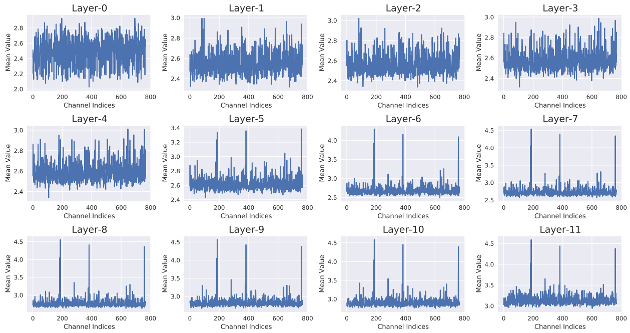


Figure 41: The plot curves of the important scores calculated on KITTI at each transformer layer.

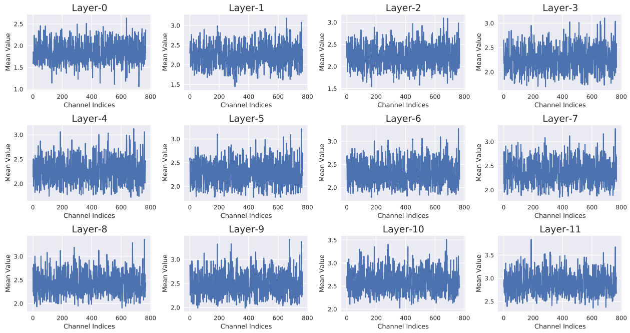


Figure 42: The plot curves of the important scores calculated on dSprites-location at each transformer layer.

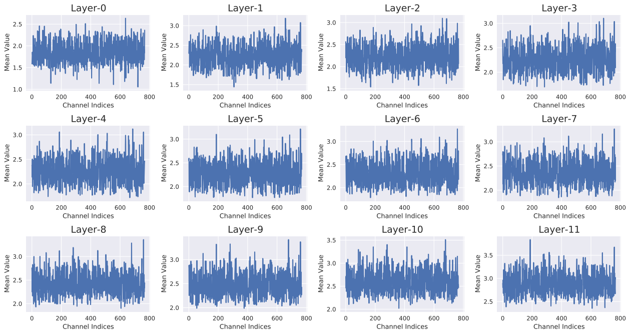


Figure 43: The plot curves of the important scores calculated on dSprites-orientation at each transformer layer.

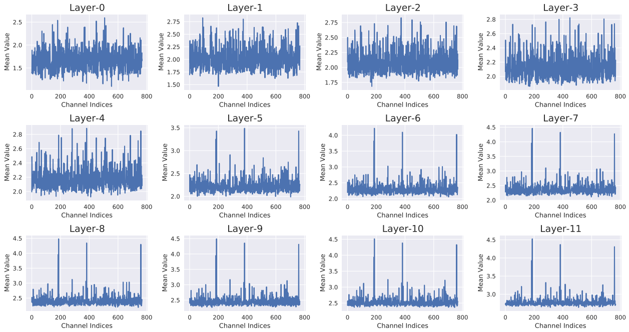


Figure 44: The plot curves of the important scores calculated on SmallNORB-azimuth at each transformer layer.

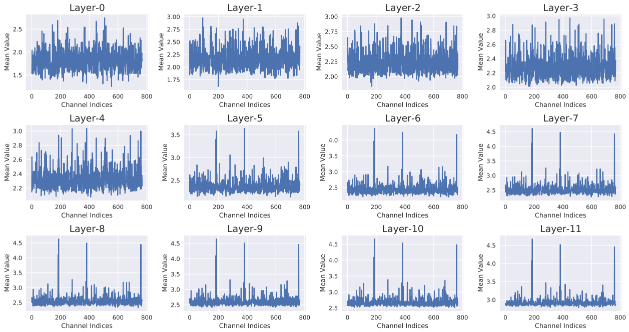


Figure 45: The plot curves of the important scores calculated on SmallNORB-elevation at each transformer layer.