Revisiting Vision Transformer from the View of Path Ensemble

Shuning Chang[1*]   Pichao Wang[2†‡]   Hao Luo[2]   Fan Wang[2]   Mike Zheng Shou[1‡]

[1]Show Lab, National University of Singapore   [2]Alibaba Group

changshuning@u.nus.edu,{michuan.lh, fan.w}@alibaba-inc.com, {pichaowang, mike.zheng.shou}@gmail.com

Abstract

Vision Transformers (ViTs) are normally regarded as a stack of transformer layers. In this work, we propose a novel view of ViTs showing that they can be seen as ensemble networks containing multiple parallel paths with different lengths. Specifically, we equivalently transform the traditional cascade of multi-head self-attention (MSA) and feed-forward network (FFN) into three parallel paths in each transformer layer. Then, we utilize the identity connection in our new transformer form and further transform the ViT into an explicit multi-path ensemble network. From the new perspective, these paths perform two functions: the first is to provide the feature for the classifier directly, and the second is to provide the lower-level feature representation for subsequent longer paths. We investigate the influence of each path for the final prediction and discover that some paths even pull down the performance. Therefore, we propose the path pruning and EnsembleScale skills for improvement, which cut out the underperforming paths and reweight the ensemble components, respectively, to optimize the path combination and make the short paths focus on providing high-quality representation for subsequent paths. We also demonstrate that our path combination strategies can help ViTs go deeper and act as high-pass filters to filter out partial low-frequency signals. To further enhance the representation of paths served for subsequent paths, self-distillation is applied to transfer knowledge from the long paths to the short paths. This work calls for more future research to explain and design ViTs from new perspectives.

1. Introduction

Vision Transformer (ViT) [15] consists of alternating layers of Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN). Most follow-ups [36, 46, 41, 43, 22, 49, 28] focus on polishing these two core modules and create various ViT variants. However, most of them do not
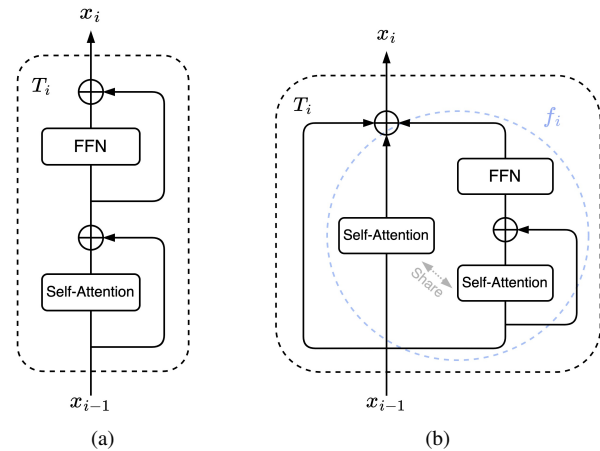

Figure 1: (a) Standard transformer form in modern ViTs is generally seen as a cascade of self-attention and FFN. (b) A three-path parallel form of transformer obtained by the equivalent transformation of (a).

break the basic ViT structure, *i.e.*, a stack of transformers containing residual sub-layers, for analysis.

Residual connections [18] are universally adopted to bypass their sub-layers in ViTs, allowing data to flow from the previous layer directly to the subsequent layer. They are defined as the form

$$x_i = g_i(x_{i-1}) + R_i(x_{i-1}), \qquad (1)$$

where the layer function g_i and R_i are typically identity and main building block. In ViTs, we observe that nearly all the non-linear structures accord with the form of R_i in Eq. 1, such as MHSA and FFN, and only linear structures exist between R_i and R_{i+1} in most cases, so that the final feature fed into classifier can be seen as a linear combination of multi-path output. This key insightful observation inspires us that the ViTs can be viewed as a collection of many paths instead of a traditional single deep network. Specifically, we equivalently transform the traditional cascade of MHSA and FFN into three parallel paths in each transformer layer, as shown in Figure 1. Then, we utilize the identity connection in our new transformer form and further transform the ViT into an explicit multi-path ensemble network. Our ensemble network is equivalent to the tradi-

tional structure, which can be verified by mathematics and experiments, while the output of each path can be operated independently.

In our ensemble view, each path performs two functions: the first is to provide the feature for the classifier directly, and the second is to provide the feature representation for subsequent longer paths. We propose new path combination and self-distillation to boost two functions separately to improve the performance of ViTs. We investigate the contribution of each path for the final prediction by analyzing the cosine distance and ablating different paths, and reveal that not all the paths are beneficial for the final results. Based on this observation, we design two simple and FLOPs-free path combination methods to optimize their combinations: path pruning which prunes underperforming paths, and EnsembleScale which re-weights the paths and makes the short paths focus on extracting high-quality representations for subsequent paths. Moreover, we discover that the model tends to enlarge the scales of the features of long paths to dilute the component of short paths, which increases the difficulty of optimization and raises the risk of divergence in deeper ViTs. Our EnsembleScale can make the model adjust the scale of EnsembleScale instead of features to alleviate this issue. According to the recent study of ViTs in frequency domain [40, 30], the low-pass filter property of self-attention weakens the expression of high-frequency signals. Our path combination methods can act as high-pass filters to remove partial useless low-frequency signals, and it achieves the goal of improving the first function of paths.

To further improve the second function, that is, improving their representation utilized by the subsequent paths, we propose to transfer knowledge between different paths by knowledge distillation (KD). Thanks to the ensemble-like structure, we can perform self-distillation in a general teacher-student knowledge distillation way. We apply two types of distillation, prediction-logit distillation and hidden-state distillation, to allow the shorter paths to mimic the logit and feature relation of longer paths. Compared with traditional self-distillation methods [52, 23], our method does not increase training parameters and memory cost.

The contributions of this paper are summarized as below.

- We propose a novel view of ViTs, which illustrates that ViTs can be seen as a collection of paths, instead of a traditional single-path network. We can improve ViTs by optimizing the paths.

- Based on the proposed view, we investigate the contribution of different paths for the final prediction and find out that not all the paths are positive. We present path pruning and EnsembleScale to boost the ensemble performance.

- To further enhance the representation ability of the paths, we design a self-distillation for ViTs. The

teacher network and student network are appropriately selected from the paths, making the knowledge transfer among the paths effectively.

2. Related work

Vision transformers. Vision transformer (ViT) [15] first introduces a pure Transformer backbone for image classification. ViT splits images into a sequence of tokens, and then adopts standard Transformer layers, consisting of Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN), to model these tokens. Transformer, the core of ViT, and its sub-layers, MHSA and FFN, are improved to suit vision tasks by subsequent research and various remarkable ViT variants are proposed [36, 46, 41, 11, 43, 22, 49, 28, 10, 47, 14, 20, 27, 7, 48]. For instance, PVT [41] incorporates a spatial-reduction attention layer to achieve a high-resolution multi-scale design, favoring dense prediction tasks under limited computational cost. CVT [43] proposes a convolutional projection in the attention layer to combine the merits of CNNs and Transformers. Swin Transformer [28] presents non-overlapping window partitions and restricts self-attention computation within windows to obtain linear computational complexity. Focal Transformer [47] adopts focal self-attention to capture fine-grained local and coarse-grained global interactions. MetaFormer [49] shows that replacing the self-attention with a spatial pooling operator can achieve competitive performance on many vision tasks and conclude the success of ViTs from the MetaFormer architecture. Those variants obey the basic ViT architecture, *i.e.*, a stack of transformers containing residual sub-layers, which is the base of our ensemble view. Nearly no non-linear structures between adjacent residual blocks ensure that the final feature can be equivalently decoupled into multiple paths.

Beyond image classification, ViT variants further inspire the application of transformer to other vision tasks, such as object detection [5, 57, 53, 12], semantic segmentation [39, 42, 54], and self-supervised learning [9, 6, 26].

Ensemble. Neural network ensemble is a learning paradigm to collect a finite number of neural networks for the same task, originating from [17]. A neural network ensemble is normally constructed in two steps, training a number of component networks and combining the component predictions. The most classical methods of training component neural networks include Bagging [4] and Boosting [35]. For combining the predictions of component neural networks, the most prevailing approaches are plurality voting or majority voting [17], simple averaging [29], and weighted averaging [32]. Zhou *et al.* [56] discuss the relationship between the ensemble and its component neural networks and uncovers that ensembling many of the components may be better than ensembling all of them. In this

paper, the traditional ViTs are seen as an ensemble view and show ensemble-like behavior. We explore the contribution of each path for the vision task and improve the performance by deleting the weak components or introducing EnsembleScale to re-weight the components. Veit *et al.* [38] show that convolutional residual networks can be interpreted as a collection of many paths. The paths in [38] are unrolled recursively from the bottom of the models, while our paths come from the linear combination of the top feature of the models. Moreover, the ensemble in [38] is not an equivalent transformation since they neglect the non-linear structures between adjacent residual blocks in CNNs, which limits their practical application.

Knowledge distillation. Knowledge distillation [19] transfers knowledge from a teacher model to a student model in a teacher-student framework. It has been widely studied in convolutional networks [34, 31, 24, 8, 16, 3]. Recently, several works develop distillation techniques for ViTs. DeiT [36] applies a distillation token to transfer the knowledge from CNNs to transformers. MiniViT [51] and TinyViT [44] adopt knowledge distillation to achieve lightweight ViTs. Manifold [21] proposes to excavate patch-level information to enhance ViT distillation.

Besides general knowledge distillation, several works try to use the student network itself as a teacher, named self-knowledge distillation. BYOT [52] boosts the low-level features by additional supervision from labels and the deepest layer. Xu *et al.* [45] adopt different data distortions to deal with the same data and reduce their feature distance from a single network. CS-KD [50] minimizes the KL divergence between predictive distributions from the same class. PS-KD [23] enhances the k-th epoch training model by incorporating the knowledge from the $k-1$-th epoch model. Although self-distillation methods avoid teacher networks, they increase other overhead, *e.g.*, extra large-scale parameters [52], memory [23], and computational cost [45, 50].

3. Method

The ensemble view of ViTs is first introduced in Sec. 3.1, followed by the proposed path combination in Sec. 3.2 and self-distillation in Sec. 3.3.

3.1. The ensemble view of ViTs

To make it easier to explain, we adopt the vanilla ViT [15] to instantiate a particular ensemble network. Consider a ViT network with N transformer blocks $[T_i]_{i=1}^N$. For transformer T_i consisting of two sub-layers, MHSA and FFN, x_{i-1} and x_i are defined as its input and output separately. Transformer is conventionally illustrated as in Figure 1a, which is a cascade of MHSA and FFN with residual connections and a natural representation is written as

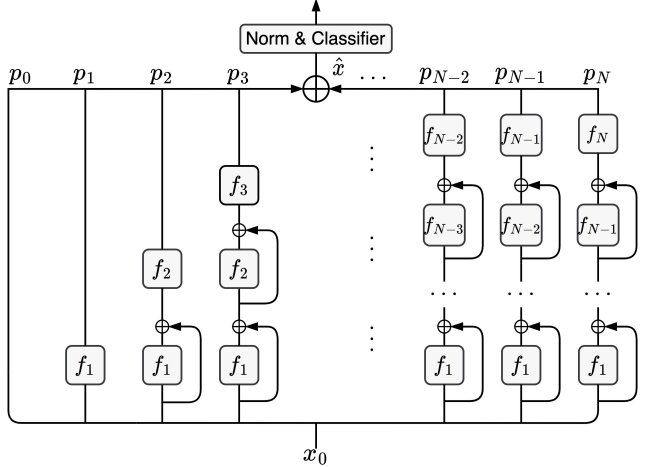$$x'_{i-1} = x_{i-1} + MHSA_i(x_{i-1}), \qquad (2)$$

Figure 2: Our ensemble view of ViTs.

$$x_i = x'_{i-1} + FFN_i(x'_{i-1}). \qquad (3)$$

In a ViT, Eq. 2 and Eq. 3 are alternately executed N times and the output of the last transformer x_N is fed into a classifier to generate the final prediction. We observe that there are two identity skip connections bridging the input and output of MHSA and FFN, respectively, which means that the input and output of MHSA can reach the tail of this transformer directly. Therefore, we can rearrange the transformer from a two-layer cascade form to a three-path parallel form as shown in Figure 1b. The three paths include an identity skip connection, an MHSA layer, and an FFN followed by an MHSA layer. Two MHSA layers are weight sharing. The mathematical expression of Figure 1b is

$$\begin{aligned} x_i =& x_{i-1} + MHSA_i(x_{i-1}) \\ &+ FFN_i(x_{i-1} + MHSA_i(x_{i-1})). \end{aligned} \qquad (4)$$

The three paths in Figure 1b correspond to the three terms in Eq. 4. In fact, Eq. 4 is equivalent to the combination of Eq. 2 and Eq. 3 by eliminating x'_{i-1}. We combine two parameterized paths into one network f_i for convenience and thus the i-th transformer can be represented by

$$x_i = x_{i-1} + f_i(x_{i-1}). \qquad (5)$$

Consider the output of the last transformer $x_N = x_{N-1} + f_N(x_{N-1})$, it can be seen as the a linear combination of x_{N-1} and $f_N(x_{N-1})$. The term x_{N-1} can be further decoupled into x_{N-2} and $f_{N-1}(x_{N-2})$. In this recursive paradigm, we can unroll the x_N into a linear number of terms, expanding one layer at each substitution step:

$$\begin{aligned} x_N &= x_{N-1} + f_N(x_{N-1}) \\ &= x_{N-2} + f_{N-1}(x_{N-2}) + f_N(x_{N-1}) \qquad (6) \\ &\cdots \\ &= x_0 + f_1(x_0) + f_2(x_1) + \cdots + f_N(x_{N-1}). \end{aligned}$$

As shown in Eq. 6, we transform the top feature x_N into a linear combination of $N + 1$ terms. The top feature x_N is used to extract the class token (or average token), and then the class token (or average token) is input into a linear classifier to obtain the final predicted result. We regard the $N+1$ terms as $N+1$ paths and the final result as the ensemble of $N + 1$ paths. Finally, a traditional ViT architecture is transformed to an ensemble view by the aforementioned transformation and analysis, and the ensemble view is illustrated in Figure 2.

We denote the path corresponding to the term $f_i(x_{i-1})$ as p_i ($i \in [0, N]$, where p_0 represents the term x_0). The network of p_i is composed of the $i - 1$ whole transformers $[T_j]_{j=1}^{i-1}$ and the parameterized sub-layers of transformer T_i, *i.e.*, $p_i = f_i(T_{i-1}(\cdots T_1(x_0)))$. Obviously, the path with greater subscript contains more layers. The original x_i can be denoted as the summation of the first i paths in our ensemble form:

$$x_i = \sum_{j=0}^{i} p_j. \tag{7}$$

We define the ensemble feature fed into the classifier as $\hat{x}$, *i.e.*, $\hat{x} = \sum_{i=0}^{N} p_i$. For a standard ViT, $\hat{x} = x_N$.

Note that the new nodes p_i in our view are not explicitly present in the traditional ViT architecture. **Our ensemble view provides a new perspective for ViTs, which involves developing ViT by optimizing the $N + 1$ paths.** We can process the paths to influence the final results. Moreover, the paths with different lengths also can be regarded as different frequency components of the final feature $\hat{x}$. Recent works [40, 30, 33, 2] have revealed the importance of frequency characteristics for ViTs. We show the Fourier analysis of paths in Figure 3a. The frequency of paths show a trend from raising to declining. Many low-frequency components are concentrated in the short paths. Adjusting the feature frequency also can be achieved by processing the paths.

The above discussion pertains to the most basic and simple ViT paradigm [15]. However, most state-of-the-art ViT models adopt a hierarchical structure to utilize multi-scale features [43, 41, 28, 10, 47, 14]. These models split transformers into multiple stages and insert downsampling layers to reduce resolution after each stage. The downsampling layer typically comprises linear layers which include a normalization layer and a linear or convolutional layer, without non-linear layers. As a result, hierarchical ViTs can also be transformed into our ensemble form. The normalization layer needs to be particularly expounded. ViTs widely employ LayerNorm [1] as the normalization layer in the downsampling layer. In our ensemble view, each path calculates the standard deviation independently, which makes the forward propagation of our ensemble view and the standard view not equivalent, unless synchronizing the

standard deviations. We conduct extensive experiments to study this issue and find that the model can adapt to asynchronous standard deviation. Additionally, the performance of asynchronous standard deviation slightly outperforms synchronous standard deviation when we train the models in the ensemble form from scratch. Therefore, we directly adopt the asynchronous standard deviation in subsequent studies. More details about our ensemble form of hierarchical ViTs are in our supplementary material.

For non-hierarchical ViTs [15, 36, 22], all the intermediate variables of our ensemble form are also computed in the standard form. Hence, the FLOPs and throughput of our ensemble form are equal to the standard form. However, for hierarchical ViTs, the downsampling layers need to be implemented in each path in our ensemble view, which incurs significant computational cost. To address this issue, we apply the strategy of "summation before downsampling". Unlike computing the output of all the paths before combining, we sum all the existing paths once encountering downsampling layers. In this way, the computational complexity from additional downsampling layers is linear to the number of downsampling layers rather than linear to the number of transformers. In the next subsection, we can see that the extra computational complexity can be reduced further. The practical FLOPs increase is negligible.

Figure 2 makes clear the processes of the well-known ViT in a novel ensemble view, where the data flow along multiple paths with different depth to form the ensemble prediction and each path also acts as the bottom network of the longer paths. Consequently, each path performs two functions: the first is to provide the feature for the classifier, and the second is to extract semantic representation for subsequent long paths. Based on these observations, we formulate the following questions: is ensembling all the paths the optimal solution? If not, how to optimize the combination of paths? Besides the classification supervision, do we have a better manner to improve the representation of the paths?

3.2. Path combination

The ensemble feature $\hat{x} = \sum_{i=0}^{N} p_i$ is fed into the linear classifier to produce prediction (the extraction of class token or average pooling is ignored for brevity). This procedure can be simplified as $y = \hat{x}w + b$, where y is the predicted result and w and b are the weight and the bias of the classifier. w can be regarded as a set of c vectors, *i.e.*, $[w_1, w_2, \cdots, w_c]$, where c is the number of classes. Assuming that the class gt corresponding to w_{gt} is the ground truth, the model is expected to make $\hat{x}w_{gt} > \hat{x}w_i$ ($i \in [1, c]$ and $i \neq gt$). From the view of high dimensional space, it is equivalent to minimize the angle between $\hat{x}$ and w_{gt}. We use $\hat{x}$ to approximately substitute w_{gt} to measure the classification ability of each path. A toy experiment is conducted
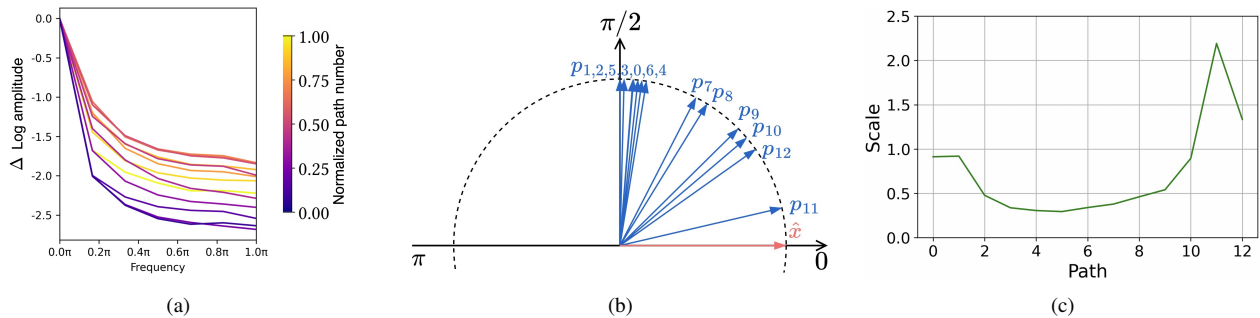
Figure 3: (a) Relative log amplitudes of Fourier transformed path features show the trend of the frequency of paths from raising to decline. This visualization refers to [30]. (b) The cosine similarity projected in the interval $[0, \pi]$ between each path and ensemble feature $\hat{x}$. (c) The scales of the paths.

by taking a ViT-S model with 12 transformer layers pre-trained on the ImageNet-1K training set. We normalize the output of the paths and project the cosine similarity between each path and $\hat{x}$ in the interval $[0, \pi]$ as shown in Figure 3b. The features from short paths are nearly orthogonal to $\hat{x}$, reflecting their weak classification ability.

Short paths with weak classification ability can play two roles in the combination of final prediction: (i) providing low-level information to revise the results of long paths; (ii) acting as the noise to be diminished by enlarging the scale of the features of long paths or sparsifying classifier weight w. We conduct another experiment to evaluate this ViT-S with different combinations of the paths on the validation set and show the results in Table 1. Note that the model is not fine-tunned even though it has been changed due to path ablation. Table 1 presents that several long paths contribute the majority of accuracy. In particular, the last three paths, *i.e.*, p_{10}, p_{11}, and p_{12} attain 99.9% of the baseline accuracy. Moreover, we notice that the combination of $[p_i]_{i=2}^{12}$ is slightly superior to the baseline even though the parameters of normalization layer and classifier are optimized for the baseline. According to these evidences, we recognize that the features of short paths do not necessarily benefit the final prediction.

We use the l_1 norm to calculate the scales of $[p_i]_{i=0}^{12}$ and plot them in Figure 3c. The scale curve presents an obvious escalating trend and reaches the peak at p_{11}, which indicates that the model fights against the noisy features by enhancing the scale of the main components. However, this way increases the difficulty of optimization and makes the model unstable with more layers. We argue that this is one of the factors causing performance saturation in deeper ViTs [37, 40].

Built on the above analysis, we propose path pruning and EnsembleScale to optimize the path combination.

Path pruning. We prune some short paths as shown in Figure 4 and force the shallow transformers to focus on

The combination of paths	Top-1 Acc (%)
Baseline ($p_{12}, p_{11}, \cdots, p_0$)	80.31*
p_{12}	78.70
p_{11}	79.70
p_{12}, p_{11}	80.14
p_{12}, p_{11}, p_{10}	80.22
$p_{12}, p_{11}, \cdots, p_2$	80.36

Table 1: Inference the pre-trained ViT-S model containing 12 transformer layers with different combinations of paths. The baseline model contains the full paths. *: Using average pooling to replace class token like [37].

extracting low-level semantic representation for subsequent layers. We restate that the paths in our ensemble view are not explicit in the traditional view of Figure 1a. Path pruning is not equivalent to removing residual connections in Figure 1a. We only prevent the ensemble prediction from combining the predictions from short paths. The shallow layers can be optimized well by residual connection in the remaining paths.

Path pruning has different effects on the computational complexity of non-hierarchical and hierarchical ViTs. Non-hierarchical ViTs remain unaffected by path pruning, whereas for hierarchical ViTs, it can help reduce the additional FLOPs required by downsampling layers. Take Swin-T [28] as an example of the latter. Swin-T needs to compute three additional downsampling layers in the ensemble form. If we remove the first three paths, the downsampling layer following stage 1 is not needed. The FLOPs of the ensemble form are equal to those of the standard form when only the last two paths are saved.

EnsembleScale. We propose EnsembleScale which is a per-channel multiplication of the vector produced by each path to adaptively re-weight the combination of the paths. Formally, EnsembleScale is a multiplication by a diagonal
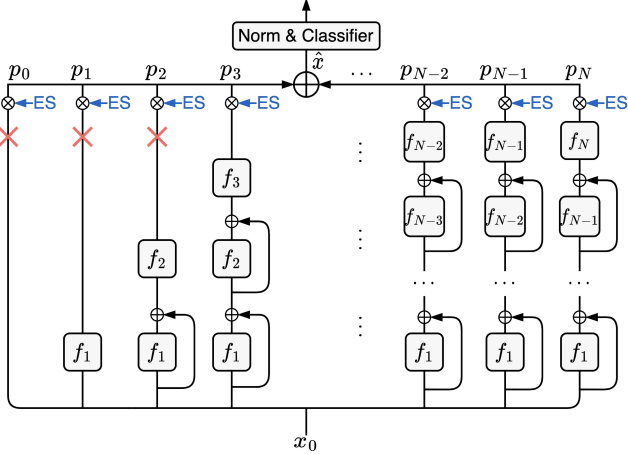
Figure 4: We propose two schemes, path pruning and EnsembleScale, to optimize path combination. × represents cutting out the corresponding path and ES is short for EnsembleScale.

matrix on the output of each path, denoted as

$$\hat{x} = \sum_{i=0}^{N} diag(\lambda_{i,1}, \cdots, \lambda_{i,d}) \times p_i, \qquad (8)$$

where the parameters $\lambda_{i,j}$ are learnable weights and d is the number of channels. We initialize the EnsembleScale progressively from 10^{-5} to 1.0 based on the analysis of short paths. EnsembleScale can be regarded as a soft path pruning. Another functionality of EnsembleScale is that the model can adjust the scale of paths by EnsembleScale rather than features in case that the scale of the feature expands with depth, which can help ViTs go deeper.

The effectiveness of our path combination strategies comes from two folds. First, the features generated by short paths focus on providing low-level representation for subsequent longer paths instead of minimizing the classification error. Second, from the perspective of frequency domain, our path combination methods mainly filter out useless low-frequency signals, amounting to amplifying the effect of high-frequency signals. Recent works [40, 30, 33, 2] validate that the high-frequency components are generally overwhelmed in ViTs and appropriately compensating for them can boost the performance.

3.3. Self-distillation

Path combination solves the combination of the paths, but it cannot actively optimize the training of the paths. To enhance their representation, we introduce self-distillation to transfer knowledge from the longer paths to the shorter paths. Two types of distillation are considered, *i.e.*, prediction-logit distillation and hidden-state distillation.

Prediction-logit distillation. Given two paths, the path with the deeper network is selected as the teacher p_t and the other as the student p_s. Then, the classifier of the overall

model is employed to extract their logits, to avoid introducing additional networks. Finally, we force p_s to imitate p_t to regularize the student path. This is achieved by a Kullback-Leibler divergence loss as below:

$$\mathcal{L}_{pl} = KL(\sigma(\frac{Cls(p_s)}{T})||\sigma(\frac{Cls(p_t)}{T})), \qquad (9)$$

where $\sigma(\cdot)$ is the Softmax function, T is a temperature value controlling the smoothness of the logits, and Cls denotes the classifier including a LayerNorm and a linear layer. We do not update the parameters of the classifier in this loss.

Hidden-state distillation. We compute the relations among tokens in p_s and p_t, respectively, and obtain two relation matrices defined by $R_i = softmax(p_i \cdot p_i^T / \sqrt{d})$. The hidden-state distillation loss based on relation matrices is achieved by another Kullback-Leibler divergence loss:

$$\mathcal{L}_{hidden} = KL(R_s||R_t). \qquad (10)$$

In experiments, we find that a large representation gap between the teacher and the student leads to inferior performance. Thus, a distance constant Δ is set to constrain teacher-student pair, *i.e.*, $p_{i+\Delta}$ teaching p_i. Δ is set to 2 by default. The final distillation objective function is formulated as

$$\mathcal{L}_{kd} = \sum_{i=s}^{N-\Delta} \alpha_i \mathcal{L}_{pl}(p_i, p_{i+\Delta}) + \beta_i \mathcal{L}_{hidden}(p_i, p_{i+\Delta}), \quad (11)$$

where s represents the subscript of the starting path, and α_i and β_i are hyperparameters to balance the loss.

4. Experiments

In this section, we report our experimental results related to path combination and self-distillation.

Experimental settings. Our method is verified on two representative ViT models, DeiT [36] and Swin [28]. All the models are trained on the ImageNet [13] with 1.28M training images and 50K validation images from 1,000 classes. The image resolution in training and inference is 224×224. All the models are trained for a total of 300 epochs, while the batch size is set to 1,024. The augmentation and regularization strategies follow the original papers of DeiT and Swin.

4.1. Path combination

Main results. Our results of path combination on DeiT and Swin are summarized in Table 2. We report top-1 accuracy, the number of parameters, and FLOPs under different path combination settings. The number of parameters and FLOPs of this paper are measured by Fvcore[1]. For DeiT, as our analysis, the components of short paths serve more as useless low-frequency information, thus pruning them can

---

[1]https://github.com/facebookresearch/fvcore

boost the performance. For instance, the performance of DeiT-S is improved by 0.4% when only keeping p_7-p_{12}. The gain comes for free without any additional parameters or FLOPs. The optimal path combination p_7-p_{12} is consistent with the cosine similarity analysis in Figure 3b which shows that p_0-p_6 are nearly orthogonal to the final ensemble feature. Our EnsembleScale re-weights the path combination and achieves better performance than path pruning in most cases.

Our methods also work well with Swin Transformers. Both path pruning and EnsembleScale can bring improvement, which demonstrates that our method is effective for diverse ViT models. The FLOPs increase compared with the baseline is from the transformation of ensemble form. Our EnsembleScale does not augment the FLOPs and path pruning can diminish this FLOPs increase due to less utilization of downsampling layers.

Making ViTs go deeper. We visualize the feature scales of the paths on DeiT-S and DeiT-S with EnsembleScale separately in Figure 5a. It can be seen that DeiT-S has to expand the scales of long paths to suppress weak features from short paths, while the model with EnsembleScale can adjust EnsembleScale (Figure 5b) to balance the weight of the paths so that the feature scales of long paths do not need to be extremely large.

We argue that large scale is one of the reasons for collapsed deep ViTs and our EnsembleScale can mitigate this issue. We experiment with more transformer layers on DeiT-S with EnsembleScale to evaluate the stability in Table 3. Note that all the hyper-parameters of "DeiT-S+ES" are the same as the vanilla DeiT-S [36]. From this table, EnsembleScale is able to converge with more layers without saturating too early. Moreover, EnsembleScale brings more improvement when more transformer layers are introduced. For example, EnsembleScale enhances 18-layer DeiT-S by 1.1%, which is far more than the 0.5% improvement it brings on 12-layer DeiT-S. Finally, these experiments support our hypothesis that it is the behavior of ViTs enlarging the scale of long paths to dilute the components of short paths that impedes ViTs going deeper. Prior works [37, 40, 55, 40] also explore the deeper ViTs but explain and solve this issue from other perspectives. We think that our work unravels a new factor of degraded deep ViTs, and EnsembleScale can actually be complementary to previous works.

Efficient dynamic ViTs. Our ensemble view can be leveraged to design efficient dynamic ViTs. We observe that many images can be predicted accurately using only a few paths, while a small fraction of difficult images require processing through the entire network. We apply a simple approach to achieve a dynamic ViT in this experiment. We use
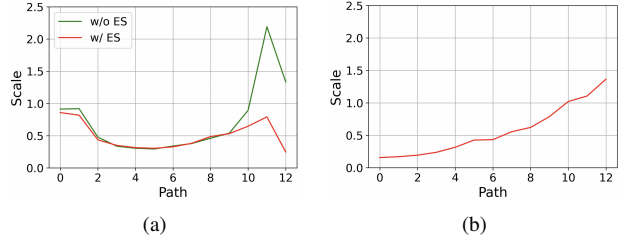

Figure 5: (a) The scales of the paths with and without EnsembleScale. (b) The scales of EnsembleScale.

two groups of EnsembleScale, denoted as ES_1 and ES_2, to combine the first seven paths and all the paths, respectively, and generate two predicted features, $\hat{x}_1$ and $\hat{x}_2$. We apply $\hat{x}_1$ to produce the initial prediction and terminate the inference process once a sufficiently confident prediction is generated. The score output by the classifier serves as the measure of confidence. Our results are shown in Table 4. After adding ES_1 and ES_2 and finetuning, the accuracy of DeiT-S increases to 80.0% when evaluating the validation set using all the paths. Subsequently, we apply our dynamic ViT to this network, resulting in a 25% reduction in FLOPs and an accuracy of 79.8% which is the same as the accuracy of the original DeiT-S. These findings indicate that our ensemble view has vast potential to achieve efficient ViT design.

The number of data processed by different stages of our dynamic ViT is shown in Table 5. It is observed that approximately 51.6% "easy" image are processed using the first 7 paths with an accuracy of 93.1%, while the remaining 48.4% "hard" data require the whole network. Our dynamic ViT is the simplest implementation. We believe that more effective and efficient dynamic ViT can be achieved based on our ensemble form.

Transfer learning. It is important to evaluate our method on other datasets with transfer learning in order to measure the generalization ability of our method. The transfer learning tasks are performed by finetuning the model on CIFAR-100 [25] and CIFAR-10 [25] as shown in Table 6. For finetuning, we use the same training setting as ImageNet-1K pre-training. Our EnsembleScale achieves superior results on both CIFAR-100 and CIFAR-10, demonstrating excellent generalization ability.

4.2. Self-distillation

Main results. We train DeiT and Swin Transformer with our self-distillation on ImageNet-1K and evaluate them on the validation set of ImageNet-1K. As shown in Table 7, both prediction-logit distillation and hidden-state distillation on DeiT and Swin can improve performance compared to the baselines, which verifies the effectiveness of our distillation methods. We also combine self-distillation with

Model	Path	ES	# Params	FLOPs	Top-1 Acc (%)
DeiT-S	-		22.05M	4.58G	79.8 (Baseline)
	p_6 - p_{12}		22.05M	4.58G	80.1
	p_7 - p_{12}		22.05M	4.58G	80.2
	p_8 - p_{12}		22.05M	4.58G	80.1
	p_0 - p_{12}	✓	22.06M	4.58G	80.3
DeiT-B	-		86.57M	17.58G	81.8 (Baseline)
	p_6 - p_{12}		86.57M	17.58G	82.0
	p_7 - p_{12}		86.57M	17.58G	82.2
	p_8 - p_{12}		86.57M	17.58G	82.2
	p_0 - p_{12}	✓	86.58M	17.58G	82.3
Swin-T	-		28.29M	4.51G	81.3 (Baseline)
	p_6 - p_{12}		28.29M	4.56G	81.5
	p_8 - p_{12}		28.29M	4.56G	81.5
	p_0 - p_{12}	✓	28.29M	4.68G	81.5
Swin-S	-		49.61M	8.77G	83.0 (Baseline)
	p_6 - p_{24}		49.61M	8.83G	83.2
	p_8 - p_{24}		49.61M	8.83G	83.2
	p_0 - p_{24}	✓	40.61M	8.95G	83.3
Swin-B	-		87.77M	15.47G	83.5 (Baseline)
	p_6 - p_{24}		87.77M	15.57G	83.7
	p_7 - p_{24}		87.77M	15.57G	83.7
	p_0 - p_{24}	✓	87.78M	15.78G	83.8

Table 2: Applying path pruning and EnsembleScale to DeiT-(Small, Base) and Swin-(Tiny, Small, Base). The top-1 accuracy, the number of parameters and FLOPs are reported under different settings. ES is short for EnsembleScale.

Depth	DeiT-S	DeiT-S + ES
18	80.1	81.2
24	78.9†	81.6

Table 3: Evaluting convergence at depth on DeiT-S. ES is short for EnsembleScale. The accuracy of DeiT is reported by [37]. †: failed before the end of training.

Model	FLOPs(G)	Accuracy(%)
DeiT-S	4.58	79.8
DeiT-S (finetuning)	4.58	80.0
Dynamic DeiT-S	3.42(-25%)	79.8

Table 4: We compare the our dynamic DeiT-S with the original DeiT-S model, reporting their respective top-1 accuracy and FLOPs.

	First 7 paths	All the paths
No. of images	25824 (51.65%)	24176 (48.35%)
Accuracy(%)	93.1	65.5

Table 5: The number of data processed by different stages of our dynamic ViT.

path combination and the performance can be further enhanced. Our path combination and self-distillation are built on our ensemble view and achieve promising results for

Model	ImageNet-1K	CIFAR-100	CIFAR-10
DeiT-S	79.8	85.9	98.1
DeiT-S with ES	80.3	87.0	98.6

Table 6: Results in transfer learning. We compare DeiT-S with EnsembleScale to DeiT-S on CIFAR-100 and CIFAR-10.

Model	Method			
	Base	PL	PL+HS	PL+HS+Path combination
DeiT-S	79.8	80.2	80.6	81.0
DeiT-B	81.8	82.1	82.4	82.7
Swin-T	81.3	81.7	82.1	82.3
Swin-S	83.0	83.4	83.6	83.8
Swin-B	83.5	84.3	84.0	84.2

Table 7: Applying our self-distillation to DeiT and Swin on ImageNet-1K. PL and HS are short for prediction-logit distillation and hidden-state distillation, respectively. We also report the results of the combination of self-distillation and path combination in the last column.

Δ	1	2	3	4
Top-1 Acc (%)	80.5	81.0	80.6	80.4

Table 8: Performance evaluation on different values of Δ on DeiT-S.

Starting path	p_0	p_2	p_4	p_5	p_6
Top-1 Acc (%)	80.6	80.8	80.8	81.0	80.8

Table 9: Performance evaluation on different starting path on DeiT-S.

nearly FLOPs-free and parameter-free, presenting the significant potential of the ensemble view.

Teacher selection. Given a path, we investigate how to select its teacher path. We set a constant Δ to control the distance between the teacher path and the student path. Different Δ are evaluated on DeiT-S in Table 8. Different from the experience in CNNs [52], employing the deepest features as teachers cannot bring improvement. Our method achieves the best performance when Δ is 2.

Do we need to distill short paths? In path combination, we halt the data flow from short paths to the final prediction. We conduct experiments to verify the effectiveness of distilling short paths. Table 9 illustrates the performance of different starting paths in Eq 11. For instance, "p_2" represents the paths p_2-p_{12} involve in distillation. We can see that the optimal result is from p_5 and distilling extremely short paths cannot improve performance.

5. Conclusion

In this paper, we revisit ViTs and propose a novel ensemble view that shows ViTs as an ensemble of multiple paths with varying lengths. We demonstrate that the transformation from the standard form to our ensemble form is equivalent, enabling us to manipulate the paths for different purposes. Through our investigation, we argue that short paths do not benefit the final prediction and propose new strategies to re-weight the paths from an ensemble learning perspective to optimize the path combination. Our method can also help ViTs go deeper and modulate frequency. Moreover, we introduce a self-distillation method to transfer knowledge from long paths to short paths to enhance the representation of the paths.

In the future, we plan to explore further ways to utilize the paths beyond the methods proposed in this paper, such as tuning the path components for downstream vision tasks. Furthermore, it would be worthwhile to investigate whether our ensemble view supports NLP networks. We hope that this work inspires more research in the future to design and optimize ViTs using an ensemble view.

Acknowledgement

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Jiawang Bai, Li Yuan, Shu-Tao Xia, Shuicheng Yan, Zhifeng Li, and Wei Liu. Improving vision transformers by revisiting high-frequency components. *arXiv preprint arXiv:2204.00993*, 2022.

[3] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10925–10934, 2022.

[4] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[5] 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*, pages 213–229. Springer, 2020.

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[7] Shuning Chang, Pichao Wang, Ming Lin, Fan Wang, David Junhao Zhang, Rong Jin, and Mike Zheng Shou. Making vision transformers efficient from a token sparsification view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6195–6205, 2023.

[8] Hanting Chen, Yunhe Wang, Chang Xu, Chao Xu, and Dacheng Tao. Learning student networks via feature embedding. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):25–35, 2020.

[9] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[10] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34:9355–9366, 2021.

[11] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

[12] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1601–1610, 2021.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[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 (CVPR)*, pages 12124–12134, June 2022.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[16] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[17] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[20] Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, 2021.

[21] Ding Jia, Kai Han, Yunhe Wang, Yehui Tang, Jianyuan Guo, Chao Zhang, and Dacheng Tao. Efficient vision transformers via fine-grained manifold distillation. *arXiv preprint arXiv:2107.01378*, 2021.

[22] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

[23] Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. Self-knowledge distillation with progressive refinement of targets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6567–6576, 2021.

[24] Nikos Komodakis and Sergey Zagoruyko. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

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

[26] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*, 2021.

[27] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv preprint arXiv:2201.04676*, 2022.

[28] 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*, pages 10012–10022, 2021.

[29] David W Opitz and Jude W Shavlik. Actively searching for an effective neural network ensemble. *Connection Science*, 8(3-4):337–354, 1996.

[30] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2022.

[31] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–284, 2018.

[32] Michael P Perrone and Leon N Cooper. When networks disagree: Ensemble methods for hybrid neural networks. Technical report, Brown Univ Providence Ri Inst for Brain and Neural Systems, 1992.

[33] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. *Advances in Neural Information Processing Systems*, 34:980–993, 2021.

[34] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[35] Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[36] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[37] 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*, pages 32–42, 2021.

[38] Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. *Advances in neural information processing systems*, 29, 2016.

[39] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5463–5474, 2021.

[40] Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. *arXiv preprint arXiv:2203.05962*, 2022.

[41] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021.

[42] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8741–8750, 2021.

[43] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021.

[44] Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Tinyvit: Fast pretraining distillation for small vision transformers. *arXiv preprint arXiv:2207.10666*, 2022.

[45] Ting-Bing Xu and Cheng-Lin Liu. Data-distortion guided self-distillation for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5565–5572, 2019.

[46] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9981–9990, October 2021.

[47] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers, 2021.

[48] Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[49] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10819–10829, 2022.

[50] Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13876–13885, 2020.

[51] Jinnian Zhang, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Minivit: Compressing vision transformers with weight multiplexing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12145–12154, 2022.

[52] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722, 2019.

[53] Minghang Zheng, Peng Gao, Renrui Zhang, Kunchang Li, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. *arXiv preprint arXiv:2011.09315*, 2020.

[54] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.

[55] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.

[56] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 137(1-2):239–263, 2002.

[57] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2020.