SELF-SLIMMING VISION TRANSFORMER

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision transformers (ViTs) have become the popular structures and outperformed convolutional neural networks (CNNs) on various vision tasks. However, such powerful transformers bring huge computation burden, due to the exhausting token-to-token comparison. To make ViTs more efficient, we can prune them from two orthogonal directions: model structure and token number. However, pruning structure decreases the model capacity and struggles to speed up ViTs. Alternatively, we observe that ViTs exhibit sparse attention with high token similarity, while reducing tokens can greatly improve the throughput. Therefore, we propose a generic self-slimming learning approach for vanilla ViTs, namely SiT. Specifically, we first design a novel Token Slimming Module (TSM), which can boost the inference efficiency of ViTs by dynamic token aggregation. Different from the token hard dropping, our TSM softly integrates redundant tokens into fewer informative ones, which can dynamically zoom visual attention without cutting off discriminative token relations in the image. Furthermore, we introduce a concise Dense Knowledge Distillation (DKD) framework, which densely transfers token information in a flexible auto-encoder manner. Due to the similar structure between teacher and student, our framework can effectively leverage both parameter and structure knowledge to accelerate training convergence. Finally, we conduct extensive experiments to evaluate our SiT. In most cases, our method can speed up ViTs by $\mathbf{3.6}\times$ while maintaining 97% of their performance. Surprisingly, by simply arming LV-ViT with our SiT, we achieve new state-of-the-art performance on ImageNet, surpassing all the CNNs and ViTs in the recent literature.

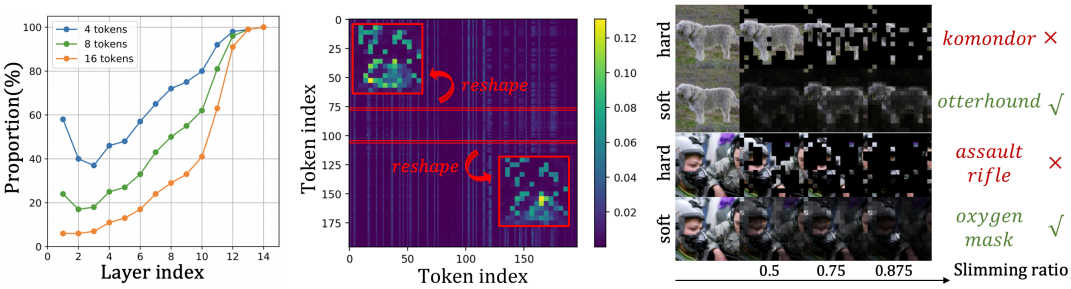
1 INTRODUCTION

Since vision transformer (ViT) (Dosovitskiy et al., 2021) started the era of transformer structure in the fundamental computer vision tasks (Carion et al., 2020; Xie et al., 2021; Chen et al., 2021b), variant transformers have been designed to challenge the dominance of convolutional neural networks (CNNs). Different from CNNs that stack convolutions to encode local features progressively, ViTs directly capture the long-term token dependencies. However, because of the exhausting token-to-token comparison, current powerful transformers require huge computation, limiting their wide application in reality (Graham et al., 2021). For this reason, we aim to design a generic learning framework for boosting the efficiency of vanilla vision transformers in this paper.

To make ViTs more efficient, we can prune them from two orthogonal directions, i.e., model structure and token number. Structure pruning has been popular in CNNs (He et al., 2017; Lin et al., 2018). However, as the model capacity will be decreased by structure pruning, it requires iterative optimization to maintain the performance. Besides, structure pruning struggles to speed up ViTs, for example, S^2ViTE (Chen et al., 2021c) can only improve the inference speed by $1.3\times$. We try to prune LV-ViT-M (Jiang et al., 2021) in different orthogonal dimensions in Table 1. It shows that token slimming can better improve the inference speed, especially for a large pruning ratio. To verify the feasibility of token slimming, we conduct a series of experiments based on LV-ViT, which reveals that sparse attention with high token similarity exists in ViTs. In Figure 1a, we calculate the correlation coefficients among tokens and count the proportion that is at least similar ($\geq$0.7) to 4/8/16 tokens in different layers. It shows that even in the first layer, more than 60% of tokens are similar to the other 3 tokens and the token similarity becomes higher in the deeper layer. Besides, the attention tends to focus on the specific tokens in the deeper layers (Figure 1b), which means the number of decision-relevant tokens becomes fewer. These observations demonstrate that only

Pruning dimension	Throughput (image/s)				
	1GFLOPs	2GFLOPs	4GFLOPs	6GFLOPs	12GFLOPs
Structure-width	2651 (3.4×)	2168 (2.8×)	1517 (2.0×)	1204 (1.6×)	774
Structure-depth	6646 (8.6×)	4709 (6.1×)	2340 (3.0×)	1623 (2.1×)	774
Token number	**10680 (13.8×)**	**5328 (6.9×)**	**2544 (3.3×)**	**1781 (2.3×)**	774

Table 1: **Throughput vs. FLOPs.** We prune LV-ViT-M in different orthogonal dimensions. It shows that token slimming achieves the highest throughput at the same FLOPs.



(a) Token similarity becomes higher in deeper layers.

(b) All tokens focus on the same informative tokens in deeper layers.

(c) Our *soft slimming* can automatically zoom the attention scope.

Figure 1: Visualizations of LV-ViT and comparison between hard dropping and soft slimming.

a few token candidates indicate meaningful information. Intuitively, we can progressively drop the redundant tokens as the network deepens. Recent studies have tried to compress tokens via data-independent dropping with minimizing reconstruction error (Tang et al., 2021), and data-dependent dropping with differentiable scoring (Rao et al., 2021). However, data-independent dropping requires layer-by-layer optimization, which is hard to generalize. Moreover, the token hard dropping will inevitably discard the vital tokens as the dropping ratio increases, e.g., the shape of the otterhound is destroyed in the deep layer (Figure 1c), thus limiting its performance.

In contrast, we propose token soft slimming to dynamically aggregate decision-relevant information into a slimming token set. Specifically, we design a concise Token Slimming Module (TSM), which generates decision-relevant tokens via a data-dependent weight matrix. As shown in Figure 1c, by simply inserting multiple TSMs in LV-ViT, our network can learn to localize the key object tokens. More importantly, the attention scope can be zoomed automatically without cutting off the discriminative token relations, e.g, our network can concentrate on the most informative parts of the otterhound and the oxygen mask, which is totally different from the token hard dropping. Undeniably, token slimming will impair the capacity of representation learning, wherein some decision-relevant information will be lost. Considering the invariant structure, the original network can teach its slimming version by a dense (layer-to-layer) supervision manner. Therefore, we introduce a novel Dense Knowledge Distillation (DKD) algorithm that elaborately utilizes the parameter knowledge and structure knowledge, in order to achieve stable and efficient model slimming optimization. With parameter knowledge, our model can converge faster, achieving 80.5% top-1 accuracy after 125 epochs, while training from scratch after 300 epochs even drops by 0.4%. To make use of structure knowledge, we first design a reverse version of the token slimming module (RTSM) to align the token number in each layer in a flexible auto-encoder manner, thus we can densely transfer the token information. Benefiting from the innate knowledge inheritance, our DKD is more suitable for teaching itself, i.e., self-slimming learning, though other CNN/Transformer teachers perform better.

Our self-slimming learning method can be easily applied to all vanilla vision transformers (SiT), e.g., DeiT (Touvron et al., 2021a), T2T-ViT (Yuan et al., 2021a), and LV-ViT (Jiang et al., 2021) etc. We conduct extensive experiments on ImageNet (Deng et al., 2009) to verify the effectiveness and efficiency. Interestingly, our method can perform better than DynamicViT (Rao et al., 2021) even only with TSM. Besides, our SiT-XS achieves 81.8% top-1 accuracy with $\mathbf{3.6}\times$ inference speed and SiT-L achieves competitive 85.6% top-1 accuracy while speed up by $\mathbf{1.7}\times$. More importantly, our SiT based on LV-ViT achieves the new state-of-the-art performance on ImageNet, surpassing all the CNNs and ViTs in the recent literature.

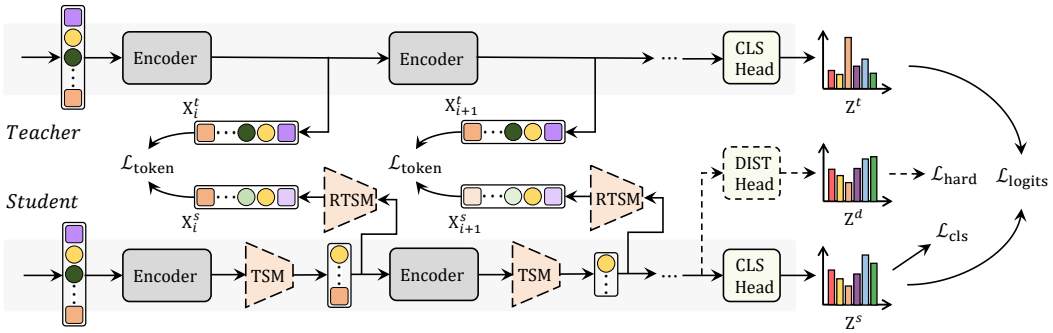


Figure 2: **The framework of our self-slimming learning.** We insert our token slimming module (TSM) and the reverse version (RTSM) into vanilla vision transformers. Specially, the RTSM is only used during training. The dense knowledge distillation (DKD) applies layer-to-layer supervision to the recovered tokens of RTSM and the final predictions. The dash lines indicate the prediction supervision from the extra CNN teacher is optional and complementary to our method.

2 Related Work

Vision Transformers. Transformer architecture (Vaswani et al., 2017a) was first proposed for machine translation in the field of natural language processing (NLP). The success in NLP inspires the application of transformer in various vision tasks, for example, DETR (Carion et al., 2020) for object detection and ViT (Dosovitskiy et al., 2021) for image recognition. ViT is the first pure transformer that achieves the state-of-the-art performance on ImageNet (Deng et al., 2009). Recent ViT variants mainly focus on better optimization and more powerful performance (Touvron et al., 2021a; Zhou et al., 2021; Touvron et al., 2021b; El-Nouby et al., 2021; Yuan et al., 2021b;a; Liu et al., 2021; Wang et al., 2021b; Jiang et al., 2021; Han et al., 2021; Chen et al., 2021a; Dong et al., 2021; Wu et al., 2021; d'Ascoli et al., 2021; Chu et al., 2021; Yang et al., 2021; Heo et al., 2021; Li et al., 2021; Guo et al., 2021). However, few of them explore to improve the efficiency of vision transformers (Graham et al., 2021). In this paper, we aim to design a general optimization framework named self-slimming to promote the efficiency of ViTs.

Transformer Slimming. The large computation of self-attention hinders the wide application of ViTs, such as detection and segmentation with the high-resolution input image. To solve this problem, several prior works concentrate on designing sparse attention (Wang et al., 2021a; Liu et al., 2021) or structure pruning (Chen et al., 2021c). S^2ViTE (Chen et al., 2021c) dynamically extracts and trains sparse subnetworks of ViTs, while sticking to a fixed small parameter budget. However, pruning model structure struggles to trim down the inference latency. Other works try to reduce the token redundancy (Rao et al., 2021; Tang et al., 2021; Xu et al., 2021) by entirely dropping the unimportant tokens. This hard dropping manner brings more improvements on throughput compared to structure pruning. Different from the above works, our SiT aggregates all tokens into fewer informative tokens in a soft manner by a concise slimming module, which can automatically zoom the attention scope to localize the key object.

3 Method

In this section, we describe our self-slimming learning for vision transformer (SiT) in detail. First, we introduce the overall architecture of SiT. Then, we explain the vital design of our SiT, i.e., token slimming module (TSM) and dense knowledge distillation (DKD). Finally, we thoroughly compare our TSM and DKD with other counterparts.

3.1 Overview of Self-Slimming Learning

In this section, we formally describe the details of our self-slimming learning for vision transformers (SiT). The overall framework is illustrated in Figure 3. We first design a lightweight Token Slimming Module (TSM) for conventional vision transformers to perform token slimming and its reverse version of token slimming module (RTSM) for token reconstruction. Following the hierarchical fea-

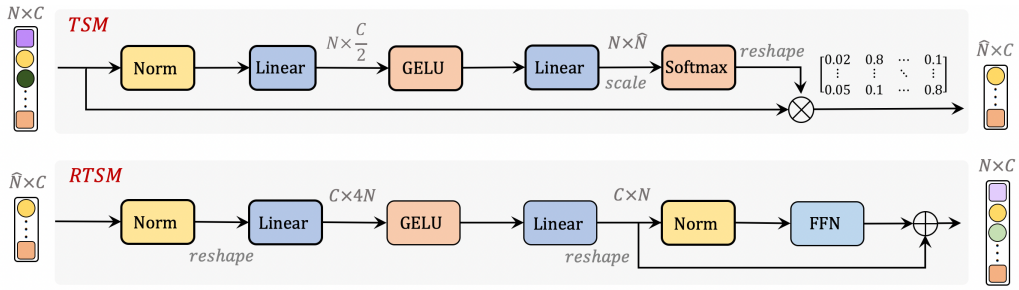


Figure 3: The pipelines of the token slimming module (TSM) and its reverse version (RTSM).

ture representations of prior works (Graham et al., 2021; Liu et al., 2021), we progressively perform token slimming three times, reducing half of the tokens every time. To decrease the inevitable information loss, we propose a layer-to-layer dense knowledge distillation (DKD), wherein the original vision transformer can serve as a teacher to minimize the difference between itself and the slimmed student. Finally, we integrate TSM and DKD to form a general self-slimming learning method for all vanilla ViTs.

3.2 Token Slimming Module

Given a sequence of N input tokens $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \cdots; \mathbf{x}_N] \in \mathbb{R}^{N \times C}$ (class token is omitted as it will never be pruned), token slimming aims to dynamically aggregate the redundant tokens to generate $\hat{N}$ informative tokens $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1; \hat{\mathbf{x}}_2; \cdots; \hat{\mathbf{x}}_{\hat{N}}]$:

$$\hat{\mathbf{X}} = \hat{\mathbf{A}}\mathbf{X}, \tag{1}$$

where $\hat{\mathbf{A}} \in \mathbb{R}^{\hat{N} \times N}$ is a normalized weight matrix:

$$\sum_{i=1}^{\hat{N}} \hat{\mathbf{A}}_{i,j} = 1, \quad where \;\; j = 1 \ldots N. \tag{2}$$

Such operation is differentiable and friendly to end-to-end training. We follow the design paradigm of self-attention (Vaswani et al., 2017b) and propose a lightweight token slimming module (TSM) shown in Figure 3:

$$\hat{\mathbf{A}} = \mathrm{Softmax}(\frac{W_q \sigma(\mathbf{X} W_k)^T}{\tau}), \tag{3}$$

where $W_k \in \mathbb{R}^{C \times \frac{C}{2}}$ and $W_q \in \mathbb{R}^{\hat{N} \times \frac{C}{2}}$ are both learnable parameters. σ and τ represents the nonlinear function (GELU) and scaling factor respectively. Similar to self-attention, TSM generates a global attention matrix, but it requires much fewer overhead in terms of throughput and memory usage during both training and inference. Thanks to the learnable scaling factor τ, the attention tends to be sparse in our experiments, which means it learns to focus on the most informative tokens.

Besides, for the followed DKD, we also design a reverse version of the token slimming module (RTSM) to reconstruct the original tokens in a flexible auto-encoder manner. Therefore, the lossless token information can be seamlessly transferred from the teacher. Note that we only perform RTSM when training, thus no extra computation is introduced during inference. We first linearly transform the informative tokens into several token candidates, thus utilizing a non-linear function to filter the vital representations. Finally, another linear transformation is performed to compress the token candidates:

$$\hat{\mathbf{X}}' = \mathbf{A}_2(\sigma(\mathbf{A}_1 \hat{\mathbf{X}})), \tag{4}$$

where $\mathbf{A}_1 \in \mathbb{R}^{4N \times \hat{N}}$ and $\mathbf{A}_2 \in \mathbb{R}^{N \times 4N}$ in our experiments. To further enhance the token representations, we introduce an extra multi-layer perceptron (MLP) block (Vaswani et al., 2017b) with skip-connection:

$$\mathbf{X}' = \hat{\mathbf{X}}' + \mathrm{MLP}(\hat{\mathbf{X}}'). \tag{5}$$

The recovered tokens $\mathbf{X}'$ will be forced to be consistent with the original tokens in DKD, which guarantees the sufficient information of the slimmed tokens $\hat{\mathbf{X}}$.

3.3 DENSE KNOWLEDGE DISTILLATION

Though token slimming significantly reduces the inference latency, it will inevitably discard some decision-relevant token candidates, leading to an unacceptable accuracy drop. To ensure the stable extraction of the decision-relevant information, we propose Dense Knowledge Distillation (DKD) that regards the original vision transformer as a teacher to provide parameter knowledge and structure knowledge. Due to the invariant model structure, we can completely load parameter knowledge from the teacher as initialization to accelerate the convergence of self-slimming learning. As for structure knowledge, we design a dense (layer-to-layer) knowledge distillation for the recovered tokens:

$$\mathcal{L}_{\text{token}} = \frac{1}{LN}\sum_{i=1}^{L}\sum_{j=1}^{N}(\mathbf{X}_{i,j}^{s} - \mathbf{X}_{i,j}^{t})^2, \tag{6}$$

where $\mathbf{X}_{i,j}^{s}$ and $\mathbf{X}_{i,j}^{t}$ refer to the j-th token embedding at the i-th layer of the student and teacher, respectively. Note that $\mathbf{X}^{s}$ refers to the recovered tokens $\mathbf{X}'$ in Eq. 5. With such dense distillation, the student model will be forced to maintain as much as knowledge in the informative tokens $\hat{\mathbf{X}}$. Besides, to alleviate the classification performance deterioration caused by token slimming, we introduce the logits distillation to minimize the predictions difference between the student and teacher:

$$\mathcal{L}_{\text{logits}} = \text{KL}(\psi(\text{Z}^{s}), \psi(\text{Z}^{t})), \tag{7}$$

where KL denotes Kullback–Leibler divergence loss and ψ is the softmax function. Z^{s} and Z^{t} are respectively the predictions of the student and teacher model. Moreover, the above DKD is complementary to the hard distillation recommended in DeiT (Touvron et al., 2021a):

$$\mathcal{L}_{\text{hard}} = \text{CrossEntropy}(\psi(\text{Z}^{d}), y^{c}), \tag{8}$$

where Z^{d} indicates the prediction of distillation head and y^{c} is the hard decision of the extra CNN teacher, which can further improve the performance with longer training epochs. Our final objective of distillation for self-slimming learning is:

$$\mathcal{L}_{\text{dist}} = \lambda_{\text{token}}\mathcal{L}_{\text{token}} + \lambda_{\text{logits}}\mathcal{L}_{\text{logits}} + \lambda_{\text{hard}}\mathcal{L}_{\text{hard}}, \tag{9}$$

where λ is the coefficient balancing the three distillation losses. We set $\lambda_{\text{logits}} = 2, \lambda_{\text{token}} = 2$ by default. λ_{hard} is set to 1 when the CNN teacher is involved. As for the training objective of self-slimming learning, we treat the classification task and the distillation task equally:

$$\mathcal{L}_{\text{cls}} = \text{CrossEntropy}(\psi(\text{Z}^{s}), \overline{y}), \tag{10}$$

$$\mathcal{L}_{\text{global}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{dist}}, \tag{11}$$

3.4 DISCUSSION

Hard dropping vs. Soft Slimming. The prior works have tried to compress tokens via hard dropping (Tang et al., 2021; Rao et al., 2021; Xu et al., 2021), in which the slimming weight $\hat{\mathbf{A}}_{i,j} \in \{0, 1\}$ is a binary decision matrix, i.e., keeping or dropping the corresponding token. However, this approach with binary decision leads to severe information loss if numerous tokens are discarded. Such weakness limits their high efficiency on ImageNet (Deng et al., 2009), wherein the objects often occupy a large part in the pictures. On the contrary, we design soft slimming with normalized weight $\hat{\mathbf{A}}_{i,j} \in (0, 1)$. It is able to discriminate the meaningful tokens in a global view, thus effectively generating decision-relevant tokens. Moreover, as shown in Figure 1c, our soft slimming can dynamically zoom the attention scope to cover the significant regions for classification.

DKD vs. Other Distillation. We compare our well-designed dense knowledge distillation with other popular distillation methods, e.g., DeiT (Touvron et al., 2021a) and LV-ViT (Jiang et al., 2021). Since the knowledge distillation often selects a stronger teacher network with different architectures, e.g., RegNet for DeiT and NFNet for LV-ViT, only the spare knowledge can be used to supervise the student, such as the single image-level or dense token-level predictions generated by the last classification layer. Due to the structural isolation between student and teacher in conventional KD, the semantic information in the intermediate layer is difficult to be utilized. In DKD, the structure is consistent between the teacher and student, it can naturally conduct densely layer-wise and token-level supervision for each layer, which further improves the stability of the model mimicking.

Model	Stage	Embed Dim	Resolution	Student Throughput (image/s)	Student Top-1 (%)	Student Top-1ϒ (%)	Teacher Throughput (image/s)	Teacher Top-1 (%)
SiT-Ti	{1,1,1,11}	320	224^2	5896 (**3.2**×)	80.1 (−2.0)	80.6 (−1.5)	1827	82.1
SiT-XS	{1,1,1,13}	384	224^2	4839 (**3.6**×)	81.1 (−2.2)	81.8 (−1.5)	1360	83.3
SiT-S	{9,3,2,2}	384	224^2	1892 (**1.4**×)	83.2 (**−0.1**)	81.8 (+0.1)	1360	83.3
SiT-M	{10,4,3,3}	512	224^2	1197 (**1.5**×)	84.1 (**−0.1**)	84.3 (+0.1)	804	84.2
SiT-L	{10,4,3,7}	768	288^2	346 (**1.7**×)	85.6 (**−0.1**)	-	204	85.7

Table 2: **Main results on ImageNet.** We apply our self-slimming learning on the state-of-the-art vanilla vision transformer LV-ViT (Jiang et al., 2021). ϒ means we adopt an extra CNN teacher. Our SiT can speed up LV-ViT **1.7**× with a slight accuracy drop. For fast inference, our SiT can maintain 97% of the performance while speeding up the original transformers **3.6**×.

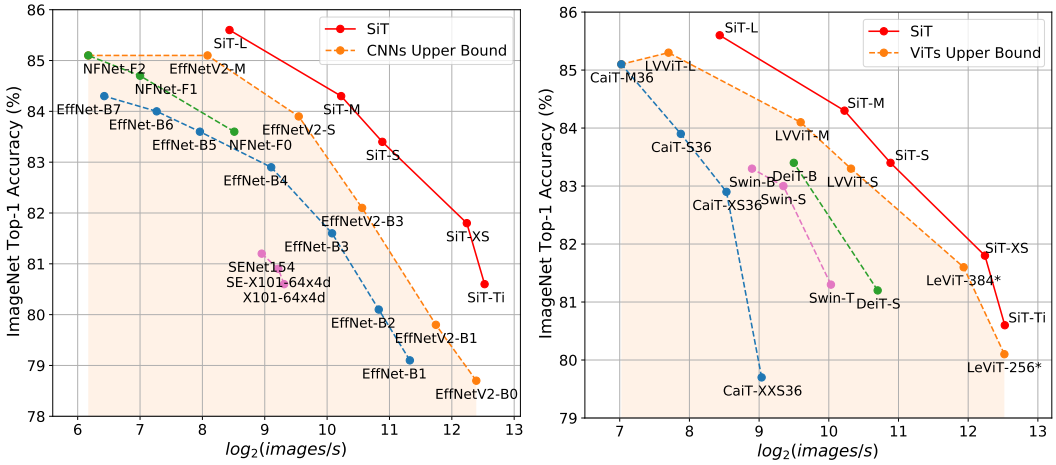


Figure 4: **Speed vs. accuracy.** "X" is short for "ResNeXt". The throughput is measured on a single 16GB V100 GPU under the same setting as Graham et al. (2021). Our SiT surpasses EfficientNetV2 (Tan & Le, 2021) and Le-ViT (Graham et al., 2021), which are designed for fast inference.

4 Experiments

4.1 Implementation Details

In this section, we conduct comprehensive experiments to empirically analyze the effectiveness of our proposed self-slimming learning for vision transformer (SiT). All the models are evaluated on the ImageNet dataset (Deng et al., 2009). For our teacher models, we train LV-ViTs (Jiang et al., 2021) following the original settings, but we replace the patch embedding module with lightweight stacked convolutions inspired by LeViT (Graham et al., 2021). All the teacher models share the same head dimension (64) and expand ratio (3) for Feed Forward Network (FFN). As for student models, all the training hyper-parameters are the same as DeiT (Touvron et al., 2021a) by defaults. For initialization, we load all the weights from the corresponding teacher models to accelerate the convergence and train them for 125 epochs. If utilizing an extra CNN teacher, we train the student for 300 epochs for better improvement. Moreover, we set different initial learning rates for the backbone and the token reconstruction branch, which are $0.0002 \times \frac{batch\ size}{1024}$ and $0.001 \times \frac{batch\ size}{1024}$ respectively. For token slimming, we insert TSM three times, thus there are four stages in SiT. The default reduction ratio $\hat{N}/N$ is set to 0.5, which means the token number is halved after slimming.

4.2 Main Results

We conduct our self-slimming learning for LV-ViT (Jiang et al., 2021), which is the state-of-the-art vanilla vision transformer. More results based on DeiT (Touvron et al., 2021a) can be found in Appendix A.1. Table 2 shows our detailed settings for different SiT variants. For SiT-Ti and SiT-XS, we explore their capacity for fast inference, thus we insert TSMs in the early layers. It demonstrates that our self-slimming method is able to speed up the original vision transformers **3.6**×, while

Model	Resolution	#Params (M)	#FLOPs (G)	Throughput (image/s)	ImageNet Top-1(%)
EfficientNet-B2 (Tan & Le, 2019)	260^2	9.1	1.1	1818	80.1
PVT-S (Wang et al., 2021b)	224^2	24.5	3.8	1017	79.8
LeViT-256* (Graham et al., 2021)	224^2	18.9	1.1	5872	80.1
CaiT-XXS36↑ (Touvron et al., 2021b)	224^2	17.3	3.8	525	79.7
SiT-Ti	224^2	15.9	1.0	**5896**	**80.6**
EfficientNet-B3 (Tan & Le, 2019)	300^2	12.2	1.9	1082	81.6
DeiT-S↑ (Touvron et al., 2021a)	224^2	22.4	4.6	1619	81.2
LeViT-384* (Graham et al., 2021)	224^2	39.1	2.4	3916	81.6
Swin-T (Liu et al., 2021)	224^2	28.3	4.5	1046	81.3
SiT-XS	224^2	25.6	1.5	**4839**	**81.8**
EfficientNet-B4 (Tan & Le, 2019)	380^2	19.3	4.6	550	82.9
DeiT-B↑ (Touvron et al., 2021a)	224^2	87.3	17.7	723	83.4
Swin-B (Liu et al., 2021)	224^2	87.8	15.5	477	83.3
LV-ViT-S (Jiang et al., 2021)	224^2	26.2	6.6	1277	83.3
SiT-S	224^2	25.6	4.0	**1892**	**83.4**
EfficientNet-B6 (Tan & Le, 2019)	528^2	43.0	19.9	154	84.0
EfficientNet-B7 (Tan & Le, 2019)	600^2	66.3	39.2	86	84.3
EfficientNetV2-S (Tan & Le, 2021)	384^2	21.5	8.5	747	83.9
NFNet-F0 (Brock et al., 2021)	256^2	71.5	12.6	365	83.6
LV-ViT-M (Jiang et al., 2021)	224^2	55.8	12.7	774	84.1
SiT-M	224^2	55.6	8.1	**1197**	**84.3**
EfficientNetV2-M (Tan & Le, 2021)	480^2	54.1	25.0	271	85.1
NFNet-F2 (Brock et al., 2021)	352^2	193.8	63.2	72	85.1
CaiT-M36↑ (Touvron et al., 2021b)	224^2	270.1	53.4	130	85.1
LV-ViT-L (Jiang et al., 2021)	288^2	150.1	58.8	208	85.3
SiT-L	288^2	148.2	34.4	**346**	**85.6**

Table 3: **Comparison to the state-of-the-art on ImageNet.** $*$ denotes the models are trained for 300 epochs for a fair comparison. Our SiT achieves the best balance between throughput and accuracy.

maintaining at least 97% of their accuracy. Besides, we adopt another CNN teacher to provide the hard label as in DeiT. The results show that the complementary prediction supervision can further improve the performance. As for other variants, we insert TSMs in the deeper layers. Surprisingly, with negligible accuracy drop, our SiTs are up to $\mathbf{1.7}\times$ faster than their teacher models. It is worth mentioning that, extra CNN prediction supervision brings little improvement, mainly because that the CNN teacher is worse than the original transformer (82.9% vs. $83.3\%/84.2\%$).

4.3 Comparison to state-of-the-art

In Table 3, we compare SiT with other competitive CNNs and ViTs. For a fair comparison, we group these methods according to their top-1 accuracy. The throughput is measured on a single 16GB V100 GPU under the same setting as LeViT (Graham et al., 2021). Our SiT-Ti is competitive with LeViT, while the throughput is $\mathbf{3.2}\times$ than that of EfficientNet (Tan & Le, 2019). Note that EfficientNet is designed via extensive neural architecture search and LeViT is elaborately designed for fast inference. For our larger model variants, they perform better than EfficientNetV2 (Tan & Le, 2021) with simple training strategies. Compared with the original LV-ViT (Jiang et al., 2021), our SiT is $\mathbf{1.5}\times$ faster than those with similar accuracy. We further visualize the comparisons to the upper bound of CNNs and ViTs in Figure 4. It clearly shows that our SiT achieves the best balance between throughput and accuracy, surpassing the recent state-of-the-art CNNs and ViTs. These results demonstrate the effectiveness and efficiency of our self-slimming method.

4.4 Ablation Studies

Does token slimming outperform structure pruning? In Table 4a, we compare token slimming with structure pruning under the same computation limit. For structure pruning, we adapt the channel and depth individually. For token slimming, we simply insert TSMs without DKD. The above models are trained from scratch for 300 epochs. We also drop tokens and train it with extra distillation as in DynamicViT (Rao et al., 2021). It shows that pruning channel achieves higher accuracy

Method	Top-1	FPS
Structure-width	76.3	2947
Structure-depth	69.4	5652
DynamicViT (Rao et al., 2021)	75.7	5762
SiT w/o DKD	**77.7**	**5896**

(a) **Efficiency comparison.** Token slimming performed by TSM yeilds the best efficiency.

Knowledge	Self	CaiT	RegNet
	83.3	83.5	82.9
None	**80.1**	79.9	79.2
Parameter	**80.5**	80.2	80.0
Parameter +Structure	**81.1**	80.6	80.2

(b) **Inherited knowledge.** Parameter knowledge accelerates convergence and structure knowledge increases accuracy.

Method	Top-1
None	79.0
T-Linear	78.8
T-Mixer	79.0
T-Linear + MLP	79.6
Our RTSM	**80.1**

(c) **Token reconstruction methods.** The MLP is critical to token reconstruction.

Method	GFLOPs	Top-1
None	3.5	82.1
AvgPool	1.0	77.4
Conv	1.0	79.3
T-Mixer	1.1	79.3
Our TSM	**1.0**	**80.1**

(d) **Token slimming methods.** The dynamic TSM reaches better accuracy than the static methods.

Method	Top-1
Baseline	77.7
$+\mathcal{L}_{\text{logits}}$	79.0$(+\mathbf{1.3})$
$+\mathcal{L}_{\text{token}}$	80.1$(+\mathbf{2.4})$
$+\mathcal{L}_{\text{hard}}$	80.2$(+\mathbf{2.5})$
+Longer training	80.6$(+\mathbf{2.9})$

(e) **Knowledge distillation.** Each distillation supervision help improve the performance.

FLOPs ratio	$\mathcal{L}_{\text{logits}}$ $+\mathcal{L}_{\text{token}}$	$\mathcal{L}_{\text{hard}}$
1	82.1	82.1
0.75	82.0	82.0
0.5	81.6	81.3
0.25	80.1	78.4

(f) **Robustness analysis.** Our self-slimming learning with DKD is robust to FLOPs ratio.

Table 4: **Ablation studies.** If not otherwise specified, all experiments for ablations are conducted on SiT-Ti and run with only 125 training epochs under the supervision of our DKD.

than pruning depth but with lower throughput. Besides, token slimming can largely improve the throughput with higher performance. However, DynamicViT performs worse than our SiT without distillation, which is mainly because token hard dropping loses much discriminative information with a large slimming ratio. Such results also demonstrate the effectiveness of our TSM.

Do parameter knowledge and structure knowledge matter to self-slimming? We further investigate whether the parameter knowledge and structure knowledge benefit the performance as shown in Table 4b. For the teacher models, we adopt different architectures (LV-ViT-S(Jiang et al., 2021), CaiT-S24(Touvron et al., 2021b), and RegNetY-16GF(Radosavovic et al., 2020)) but similar accuracies for a fair comparison. It shows that training with parameter knowledge for 125 epochs converges to higher results than those trained for 300 epochs without parameter knowledge. Moreover, we utilize structure knowledge via layer-to-layer mimicking, which can further boost the performance. It also reveals that higher similarity between students and teachers can bring greater improvements.

Dynamic vs. Static: Which aggregation manner works better for token slimming? To explore whether dynamic aggregation is better for token slimming, we perform ablation experiments as shown in Table 4d. For static aggregation, we choose different data-independent operations and maintain similar computation: 3×3 average pooling/convolution with stride 2×2, and double linear layers with GELU function ("T-Mixer"). It shows that learnable parameters are vital for token slimming since average pooling leads to a severe accuracy drop. Besides, the static aggregation methods with data-independent weights yield similar but inferior performance to our TSM (79.3% vs. 80.1%). Such comparisons prove that our TSM can generate more informative tokens.

How much does MLP bring for token reconstruction? We first reconstruct the original tokens by only single and double linear layers. As shown in Table 4c. "T-Linear" and "T-Mixer" do not bring any accuracy gains and even hurts the capacity compared with the baseline (without layer-to-layer mimicking). Surprisingly, simply introducing an MLP (Dosovitskiy et al., 2021) obviously improves the performance by 0.8% and 1.1% respectively. It shows that via enhancing the token representations individually, MLP can guarantee the sufficient information of the slimmed tokens.

Does each distillation supervision helps? Table 4e presents that the soft logits surpervision $\mathcal{L}_{\text{logits}}$ brings 1.4% top-1 accuracy gain. When further introducing layer-to-layer knowledge supervision, our model improves the accuracy by 1.1%. Finally, combining complementary hard label supervision, the top-1 accuracy reaches 80.6% with longer training epochs.

Is self-slimming learning robust to different FLOPs ratios? In Table 4f, we empirically training models with different FLOPs ratios. When the ratio is large than 0.5, our DKD and CNN distillation are both helpful for maintaining performance. However, when the ratio is small, CNN distillation leads to a higher performance drop, while our DKD only drops the accuracy by 2.0%. These results demonstrate that our self-slimming learning with DKD is robust to different FLOPs ratios.

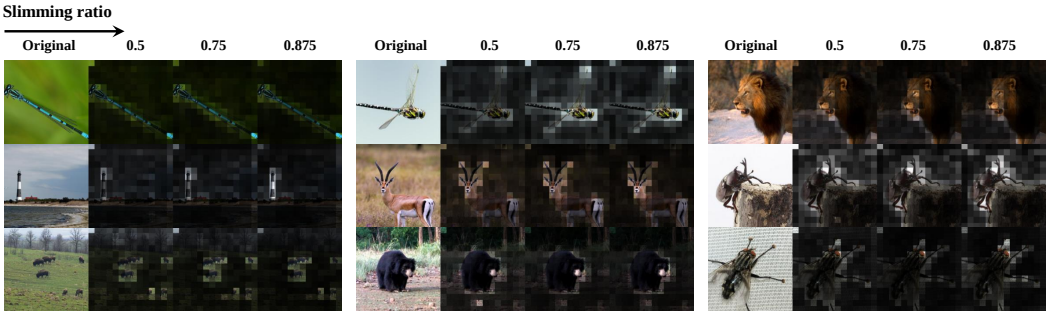


Figure 5: Visualizations of our progressive token slimming. The darker tokens contribute less to the final informative tokens. Our method can zoom the attention scope to cover the key object.

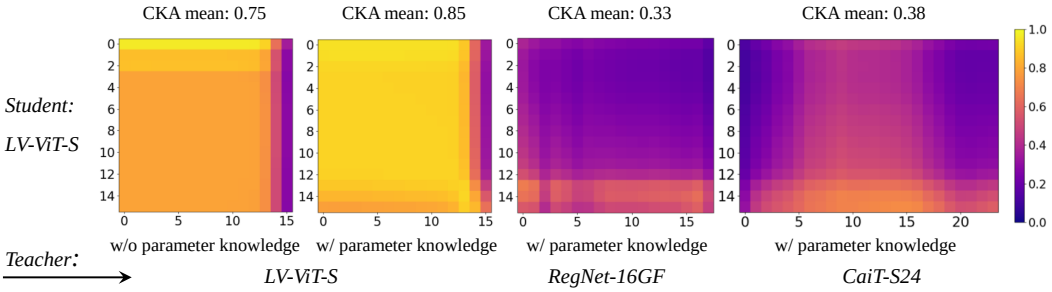


Figure 6: Cross CKA heatmap between different student models and the teacher models.

4.5 Visualization

Token slimming visualization. Figure 5 shows the original images and the token slimming procedure of our SiT-Ti. We observe that the tokens of higher scores, i.e., brighter tokens, are concentrated and tend to cover the key objects in the image. It demonstrates that our proposed TSM is able to localize the significant regions and predict accurate scores for the most informative tokens.

Model similarity visualization. In Figure 6, we compute the CKA (Kornblith et al., 2019) heatmap by comparing all layers of the student models (LV-ViT-S) with all layers of their teacher models. It shows that the CKA similarities between the similar structures are generally higher than those between different structures (0.75/0.85 vs. 0.33/0.38). Interestingly, we find the parameter knowledge inherited by the student force itself to be similar to its teacher. Besides, for similar structures, the CKA similarities in the shallow layers are higher than those in deep layers. It is mainly because we slim a large number of tokens after the third layer, leading to an inevitable information loss. As for different structures, the CKA similarities in the deep layers are higher than those in shallow layers, which is mainly because the logits distillation provides direct supervision for features in the deeper layers. Note that the above observations are consistent with the results in Table 4b, which reveals that teachers with similar structures can transfer structure knowledge better for higher performance.

5 Conclusion

In this paper, we propose a generic self-slimming learning method for vanilla vision transformers (SiT), which can speed up the ViTs with negligible accuracy drop. Our concise TSM softly integrates redundant tokens into fewer informative ones. For stable and efficient training, we introduce a novel DKD framework to leverage parameter knowledge and structure knowledge, which can densely transfer token information in a flexible auto-encoder manner. Extensive experiments demonstrate the effectiveness of our SiT. By simply arming LV-ViT with our SiT, we achieve new state-of-the-art performance on ImageNet, surpassing all the other CNNs and ViTs.

REFERENCES

Andrew Brock, Soham De, Samuel L. Smith, and K. Simonyan. High-performance large-scale image recognition without normalization. *ArXiv*, abs/2102.06171, 2021.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ArXiv*, abs/2005.12872, 2020.

Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. *ArXiv*, abs/2103.14899, 2021a.

Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12299–12310, 2021b.

Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *ArXiv*, abs/2106.04533, 2021c.

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. 2021.

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 *ICML*, 2021.

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*, pp. 248–255. Ieee, 2009.

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. *ArXiv*, abs/2107.00652, 2021.

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. *ArXiv*, abs/2010.11929, 2021.

Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, M. Douze, Armand Joulin, I. Laptev, N. Neverova, Gabriel Synnaeve, Jakob Verbeek, and H. Jégou. Xcit: Cross-covariance image transformers. *ArXiv*, abs/2106.09681, 2021.

Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Herv'e J'egou, and M. Douze. Levit: a vision transformer in convnet's clothing for faster inference. *ArXiv*, abs/2104.01136, 2021.

Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Cmt: Convolutional neural networks meet vision transformers. *ArXiv*, abs/2107.06263, 2021.

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *ArXiv*, abs/2103.00112, 2021.

Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 1389–1397, 2017.

Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. *ArXiv*, abs/2103.16302, 2021.

Zihang 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. 2021.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. *ArXiv*, abs/1905.00414, 2019.

Yawei Li, K. Zhang, Jie Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *ArXiv*, abs/2104.05707, 2021.

Shaohui Lin, Rongrong Ji, Yuchao Li, Yongjian Wu, Feiyue Huang, and Baochang Zhang. Accelerating convolutional networks via global & dynamic filter pruning. In *IJCAI*, volume 2, pp. 8, 2018.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *ArXiv*, abs/2103.14030, 2021.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross B. Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10425–10433, 2020.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *arXiv preprint arXiv:2106.02034*, 2021.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ArXiv*, abs/1905.11946, 2019.

Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training. *ArXiv*, abs/2104.00298, 2021.

Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. Patch slimming for efficient vision transformers. *arXiv preprint arXiv:2106.02852*, 2021.

Hugo Touvron, M. Cord, M. Douze, Francisco Massa, Alexandre Sablayrolles, and Herv'e J'egou. Training data-efficient image transformers & distillation through attention. *ArXiv*, abs/2012.12877, 2021a.

Hugo Touvron, M. Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Herv'e J'egou. Going deeper with image transformers. *ArXiv*, abs/2103.17239, 2021b.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017a.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017b.

Pichao Wang, Xue Wang, F. Wang, Ming Lin, Shuning Chang, Wen Xie, Hao Li, and Rong Jin. Kvt: k-nn attention for boosting vision transformers. *ArXiv*, abs/2106.00515, 2021a.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, P. Luo, and L. Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *ArXiv*, abs/2102.12122, 2021b.

Haiping Wu, Bin Xiao, Noel C. F. Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *ArXiv*, abs/2103.15808, 2021.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021.

Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. *arXiv preprint arXiv:2108.01390*, 2021.

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. *ArXiv*, abs/2107.00641, 2021.

Li Yuan, Y. Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis E. H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *ArXiv*, abs/2101.11986, 2021a.

Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *ArXiv*, abs/2106.13112, 2021b.

Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *ArXiv*, abs/2103.11886, 2021.

A ADDITIONAL RESULTS

A.1 MORE RESULTS ON DEIT

We also verify the effectiveness of our self-slimming learning on DeiT as illustrated in Table 5. For the FLOPs ratio of 0.5 and 0.25, the stage numbers are $\{3,4,3,2\}$ and $\{1,1,1,9\}$ respectively. Specifically, we conduct the experiments on the original DeiT (Touvron et al., 2021a) and its variant with lightweight convolutional patch embedding. Both models achieve similar accuracy with the same computational costs. However, we observe the performance of their students is quite different especially at a small FLOPs ratio. DeiT$_P$ suffers severe performance deterioration when 75% computation is reduced, while DeiT$_C$ only drops the accuracy by 2.5%. More importantly, DeiT$_C$ generally obtain higher accuracies than DeiT$_P$ at a relatively higher FLOPs ratio. It demonstrates that the models with convolutional patch embedding are more redundant and friendly to slimming. In addition, we also compare our DKD with the CNN distillation under different settings. The layer-to-layer dense knowledge distillation consistently brings more performance gains than CNN distillation. It is worth mentioning that, self-slimming is also complementary to the extra CNN distillation. Surprisingly, the best student model of DeiT$_C$ even outperforms the teacher by 0.6% top-1 accuracy while running $2\times$ faster under the joint supervision. These results prove the effectiveness and generalization ability of our self-slimming learning.

A.2 COMPARISONS TO DYNAMICVIT

As described in Table 6, we further compare our self-slimming learning with the recent method, i.e., DynamicViT. We observe that our SiT runs slightly faster than DynamicViT with the same FLOPs, which reveals our TSM presents better inference efficiency than the prediction module of DynamicViT. More importantly, thanks to the soft-slimming designs, SiT outperforms DynamicViT by a large margin (5.3%-10.0%) at the FLOPs ratio of 0.25. For the large FLOPs ratio, our SiT still obtains at least 0.7% higher accuracy than DynamicViT, proving the soft slimming triumphs the hard dropping manner.

A.3 VISUALIZATION

We present more visualizations of our progressive token slimming in Figure 7.

Model	FLOPs ratio	#FLOPs (G)	$\mathcal{L}_{\text{logits}}$ +$\mathcal{L}_{\text{token}}$	$\mathcal{L}_{\text{hard}}$	Throughput (image/s)	ImageNet Top-1(%)
DeiT$_P$-S	0.25	1.1	✗	✗	6413(**3.9**×)	71.6(-8.2)
		1.1	✔	✗	6413(**3.9**×)	75.9(-3.9)
		1.1	✗	✔	6286(**3.8**×)	72.9(-6.9)
		1.1	✔	✔	6286(**3.8**×)	75.3(-4.5)
	0.5	2.3	✗	✗	3308(**2.0**×)	78.6(**−1.3**)
		2.3	✔	✗	3308(**2.0**×)	79.3(**−0.5**)
		2.3	✗	✔	3262(**2.0**×)	78.8(**−1.0**)
		2.3	✔	✔	3262(**2.0**×)	79.8(**+0.0**)
	1	4.6	✗	✗	1637	79.8
DeiT$_C$-S	0.25	1.1	✗	✗	5898(**3.7**×)	76.1(-3.9)
		1.1	✔	✗	5898(**3.7**×)	78.4(-1.6)
		1.1	✗	✔	5830(**3.7**×)	77.5(-2.5)
		1.1	✔	✔	5830(**3.7**×)	78.8(-1.2)
	0.5	2.3	✗	✗	3150(**2.0**×)	79.1(**−0.9**)
		2.3	✔	✗	3150(**2.0**×)	79.9(**−0.1**)
		2.3	✗	✔	3106(**1.9**×)	80.3(**+0.3**)
		2.3	✔	✔	3106(**1.9**×)	80.6(**+0.6**)
	1	4.6	✗	✗	1597	80.0

Table 5: More results on DeiT. "DeiT$_P$" indicates the original DeiT and "DeiT$_C$" refers to the variant with lightweight convolutional patch embedding stacked by four 3×3 convolutions (2×2 stride) and one point-wise convolution.

Model	FLOPs ratio	#FLOPs (G)	DynamicViT		SiT	
			Throughput (image/s)	ImageNet Top-1(%)	Throughput (image/s)	ImageNet Top-1(%)
DeiT$_P$-S	0.25	1.1	6254(**3.8**×)	65.6(-14.2)	**6413**(**3.9**×)	**75.9**(**−3.9**)
	0.5	2.3	3248(**2.0**×)	78.4(-1.4)	**3308**(**2.0**×)	**79.3**(**−0.5**)
	1	4.6	1637	79.8	1637	79.8
DeiT$_C$-S	0.25	1.1	5689(**3.6**×)	73.4(-6.6)	**5898**(**3.7**×)	**78.4**(**−1.6**)
	0.5	2.3	3092(**1.9**×)	79.2(-0.8)	**3150**(**2.0**×)	**79.9**(**−0.1**)
	1	4.6	1597	80.0	1597	80.0

Table 6: Comparisons between DynamicViT and our SiT on DeiT.

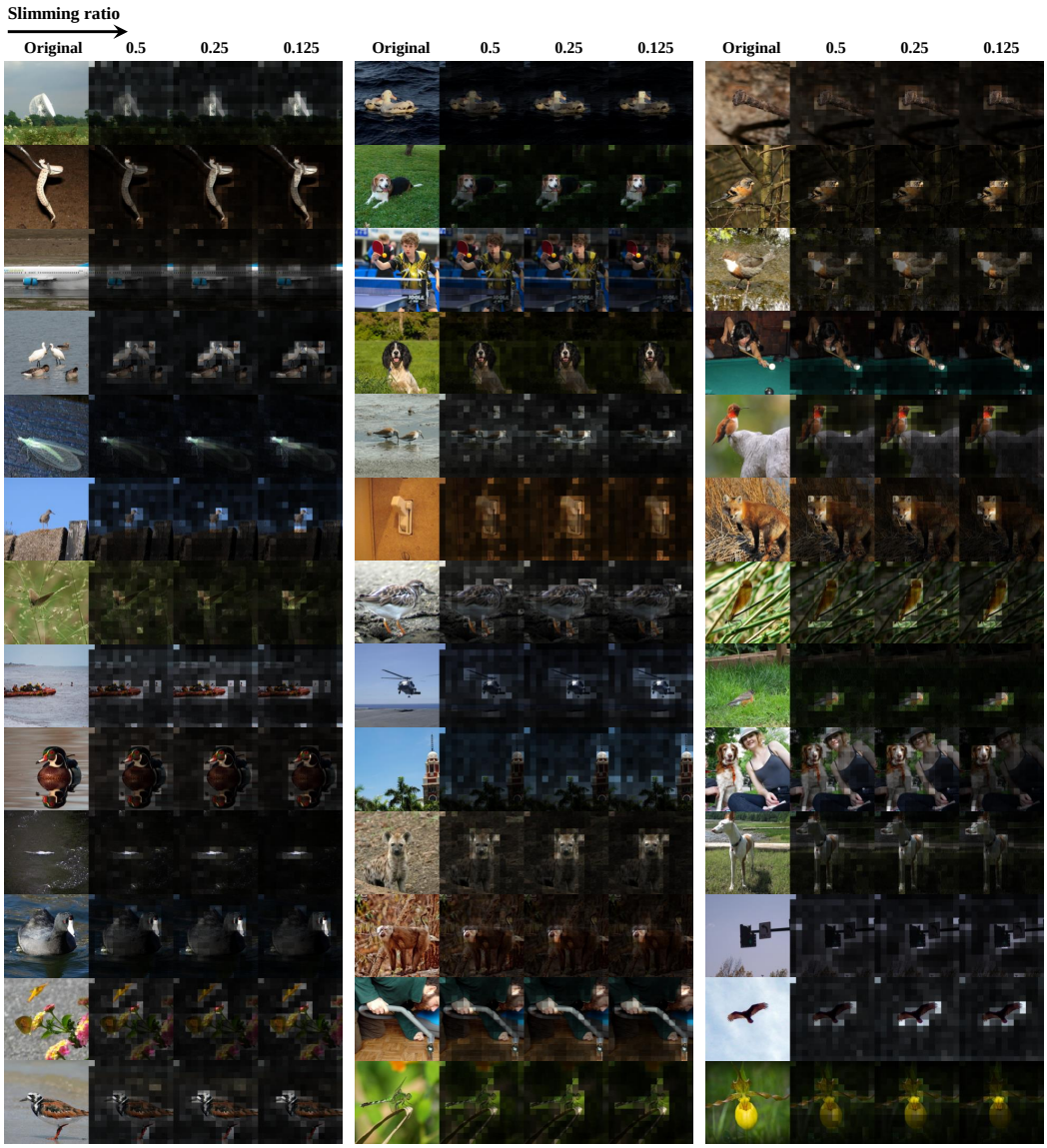


Figure 7: More visualizations of our SiT.