Making Vision Transformers Efficient from A Token Sparsification View

Shuning Chang[1*] Pichao Wang[2†‡] Ming Lin[2‡] Fan Wang[2] David Junhao Zhang[1]
Rong Jin[2] Mike Zheng Shou[1†]
[1]Show Lab, National University of Singapore [2]Alibaba Group

Abstract

The quadratic computational complexity to the number of tokens limits the practical applications of Vision Transformers (ViTs). Several works propose to prune redundant tokens to achieve efficient ViTs. However, these methods generally suffer from (i) dramatic accuracy drops, (ii) application difficulty in the local vision transformer, and (iii) non-general-purpose networks for downstream tasks. In this work, we propose a novel Semantic Token ViT (STViT), for efficient global and local vision transformers, which can also be revised to serve as backbone for downstream tasks. The semantic tokens represent cluster centers, and they are initialized by pooling image tokens in space and recovered by attention, which can adaptively represent global or local semantic information. Due to the cluster properties, a few semantic tokens can attain the same effect as vast image tokens, for both global and local vision transformers. For instance, only 16 semantic tokens on DeiT-(Tiny,Small,Base) can achieve the same accuracy with more than 100% inference speed improvement and nearly 60% FLOPs reduction; on Swin-(Tiny,Small,Base), we can employ 16 semantic tokens in each window to further speed it up by around 20% with slight accuracy increase. Besides great success in image classification, we also extend our method to video recognition. In addition, we design a STViT-R(ecovery) network to restore the detailed spatial information based on the STViT, making it work for downstream tasks, which is powerless for previous token sparsification methods. Experiments demonstrate that our method can achieve competitive results compared to the original networks in object detection and instance segmentation, with over 30% FLOPs reduction for backbone.

1. Introduction

In contrast to standard Convolutional Neural Networks (CNNs) approaches which process images pixel-by-pixel, Vision Transformers (ViTs) [15, 26, 35, 36, 43] treat an image as a sequence of patch/image tokens, and have shown promising performance in prevalent visual recognition scenarios. However, these superior performances do not come for free: the quadratic computational complexity to the number of image tokens limits their application in practice. Previous works [33, 56] have illustrated the large amount of redundancy in the image tokens and also shown the effect of filtering out unimportant tokens normally according to predefined scoring mechanism. However, these methods face the following challenges. Firstly, the predefined scoring mechanisms for filtering are generally imprecise. In Figure 1, on the left we visualize the class token values in different layers which are commonly used to score the token importance [16, 24, 45]. Different layers have different value distributions, thus using these imprecise scores for filtering would lead to unsatisfactory performance. For example, EViT [24] has an accuracy drop of 1.3% when saving 50% FLOPs on DeiT-S [35]. Secondly, the remaining tokens do not distribute evenly in space any more, making them hard to work in local vision transformers[1]. Finally, large-scale token pruning tremendously damages the spatial structure and positional information, and causes difficulties when applied to downstream tasks, which they do not propose a solution to deal with.

To solve these problems, we propose Semantic Token ViT (STViT), for efficient global and local vision transformers, which also can be revised to serve as backbone for downstream tasks. The proposed approach is based on the following observations: (i) unlike local CNNs which learn spatial structure of images, vision transformer discretizes feature map as tokens for global feature exploration, relieving the requirements for maintaining the whole image structure and information; (ii) discrete tokens are more beneficial for optimization [38]; (iii) in Figure 1, on the right shows the attention maps in different transformer layers, and there are only several vertical lines in the deep layers, which means that only a few tokens with global se-

*Work done during an internship at Alibaba Group.
†Equal corresponding authors.
‡Work done at Alibaba Group, and now affiliated with Amazon.

[1]In this paper, we define the vision transformer with global self-attention (like DeiT) as global vision transformer and the vision transformer with local self-attention (like Swin) as local vision transformer.

Figure 1. Left: the attention values of class tokens (normalized and reshaped in image shape) in different self-attention layers. Right: the attention maps in different self-attention layers. Zoom-in for better visibility.

mantic information matter. Thus, we argue that it is not necessary to maintain massive structured tokens for ViTs, especially in the deep layers. Employing a few discrete tokens with high-level semantic information can potentially achieve both high performance and efficiency.

In STViT, the semantic tokens represent the cluster centers, and the number of them is far less than the original image tokens, significantly reducing the computational cost. Inspired by the fact that multi-head attention can conduct the cluster center recovery (Supplementary A.7), we only employ the off-the-shelf self-attention to generate the semantic tokens. Specifically, the first few transformer layers are kept unchanged to obtain the image tokens with low-level features. The image tokens are then fed into our semantic token generation module (STGM) consisting of at least two transformer layers to generate semantic tokens. In each self-attention layer, the semantic tokens are input as queries, and the image tokens are fed as keys and values. The semantic tokens dynamically aggregate image tokens through the attention layers to recover cluster centers. In the first attention layer, the semantic tokens are initialized by an intra and inter-window spatial pooling which takes into account incorporating semantic information in each window and maximizing distance between adjacent windows. Thanks to this spatial initialization, the semantic tokens mainly incorporate local semantic information and achieve discrete and uniform distribution in space. In the following attention layer, besides further clustering, the semantic tokens are equipped with global cluster centers, and the network can adaptively select partial semantic tokens to focus on global semantic information. After the STGM, the original image tokens are discarded, and only semantic tokens are kept for the subsequent transformer layers. Because the generation of semantic tokens is flexible and space-aware, our method can be plugged into both global and local vision transformers. The semantic tokens can be produced in each window for the local vision transformer.

Another property of STViT is its capability to serve as a backbone for downstream tasks, such as object detection and instance segmentation. Discussions have been missing in previous methods [16, 24, 32, 45, 56] about how to use them in downstream task under the massive loss of spatial information during the token sparsification process, which actually seriously impedes the application of their method. Instead, we design a novel STViT-R network based on STViT where a recovery module and dumbbell unit are adopted to periodically restore the full resolution feature map while the intermediate transformer layers continue to use semantic tokens to save computation cost, making our method work in downstream task.

The effectiveness of the proposed method is validated via a comprehensive empirical study on image and video ViT models. Only 16 semantic tokens on DeiT-(Tiny, Small, Base) achieve nearly 50% inference time reduction without any accuracy degradation; on Swin-(Tiny, Small, Base), we also improve the inference throughput by nearly 20% with slight accuracy increase. Moreover, the proposed STViT-R achieves promising results on object detection and instance segmentation. To the best of our knowledge, this is one of first works to apply the token sparsification algorithm in local vision transformers, and use the ViTs as backbones in downstream tasks after large-scale token pruning. Our findings in ViTs uncover that maintaining the full-size feature map is unnecessary, and a few tokens with high-level semantic representations can achieve both high performance and efficiency. Thanks to its simplicity and general-purpose ability, our method can also serve as a new efficient ViT baseline architecture and a starting point for further research from the token sparsification perspective.

2. Related work

Vision transformers. Vision Transformer (ViT) [15] first introduces a pure Transformer backbone for image classification. ViT variants further inspire the applications of transformer to various vision tasks beyond image/video classification [1, 3, 20, 35, 36, 39, 43, 44, 49, 50], such as object detection [5, 12, 52, 55], semantic segmentation [37, 42, 53], and self-supervised learning [6, 9, 22]. Vanilla transformers have high computational and memory costs because the multi-head self-attention has quadratic computational complexity to the number of image tokens. Recently, various efficient ViTs have been proposed to alleviate this issue. The existing methods mainly focus on reducing the complexity of self-attention or reducing the number of tokens. Swin Transformer [26] adopts local self-attention, *i.e.*, attending neighboring tokens within a constant window size, achieving a linear computational complexity in the self-attention with high performance. Many subsequent works [11, 14, 18, 40, 46, 48, 54] follow the local

self-attention design to develop variants. Token sparsification [8, 16, 21, 24, 29–34, 41, 45, 47, 56] also attracts increasing attention.

Token sparsification. Token sparsification methods can be mainly categorized into hard pruning [8, 16, 21, 24, 29–31, 33, 34, 45, 47] and soft pruning [32, 56]. Hard pruning methods filter out some unimportant tokens according to a predefined scoring mechanism. DynamicViT [31], SPViT [21], and AdaViT [29] introduce additional prediction networks to score the tokens. Evo-ViT [45], ATS [16], and EViT [24] utilize the values of class tokens to evaluate the importance of tokens. However, it is difficult to achieve precise scoring as shown in the left of Figure 1. Therefore, they usually suffer from a significant accuracy drop. For instance, EViT [24] has an accuracy drop of 1.3% when saving 50% FLOPs on DeiT-S. Soft pruning methods generate new tokens from image tokens by importing additional attention networks. TokenLearner [32] also argue for a few tokens to replace image tokens. However, its price is a 1.8% accuracy drop when reducing 44% FLOPs, which is far inferior to concurrent works. Besides performance degradation, previous methods also have the following disadvantages. First, whether or how to extend the methods to local vision transformers remains unexplored. Second, it has not been discussed about how to serve the downstream tasks like object detection and instance segmentation after the tokens are pruned.

In our method, we apply the off-the-shelf transformer layers to reduce token number. [2, 10, 19, 23, 28, 51] adopt similar approaches to achieve efficient non-local relationships. Our method is different from them as below: (i) our method extracts local semantic information instead of non-local relationships; (ii) the semantic tokens are a few cluster centers, which can replace the massive image tokens to achieve image classification; (iii) our method specializes in pruning tokens.

3. Method

The proposed STViT is presented in this section, which aims to construct an efficient and high-performance ViT. STViT is first introduced in Section 3.1, followed by how to apply STViT in the local vision transformer in Section 3.2. Based on STViT, STViT-R is developed to restore the spatial resolution for downstream tasks in Section 3.3.

3.1. STViT

Overall architecture. An overview of STViT architecture is presented in Figure 2a. The patch embedding layer and shallow transformer layers are kept unchanged as a base module in our method. The base module copes with all the image tokens $X \in \mathbb{R}^{N_i \times C}$ to extract low-level features, where N_i is the number of image tokens and C is the number of channels. The image tokens are fed into the semantic token generation module (STGM) to generate N_s semantic tokens $S \in \mathbb{R}^{N_s \times C}$. After the STGM, the image tokens X can be discarded, and only semantic tokens S with high-level semantic information are used in all the subsequent transformers. Due to $N_s \ll N_i$, our method can significantly reduce the computational cost.

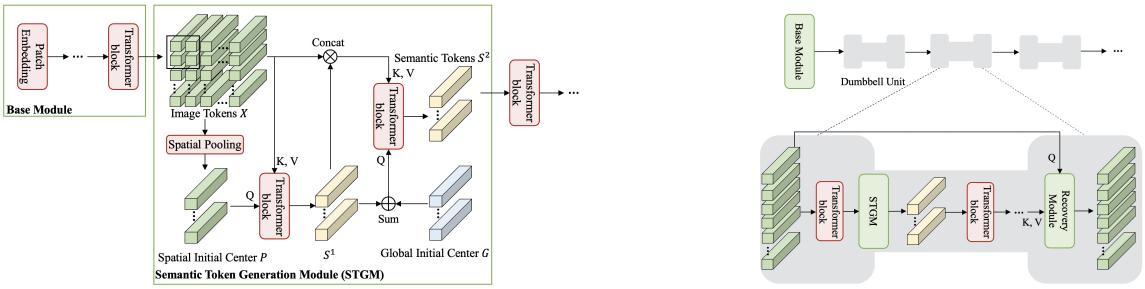
Semantic token generation module (STGM). The whole image is represented by a few tokens with high-level semantic information through clustering. Inspired by the fact that self-attention can conduct cluster center recovery (Supplementary A.7), we adopt the off-the-shelf self-attention layers to produce the semantic tokens. The STGM consists of at least two transformer layers.

The initial cluster centers $P \in \mathbb{R}^{N_s \times C}$ are generated by an spatial pooling which pools the image tokens into fixed $w_s \times w_s$ tokens, with $N_s = w_s \times w_s$. w_s is generally set as 4. The spatial pooling can be achieved by a non-parameterized adaptive spatial pooling or a super lightweight network with higher performance, intra and inter-window spatial pooling. The intentions of spatial pooling initialization are three folds. First, the initial cluster centers can distribute uniformly in space, making the generated semantic tokens more discrete and preventing the semantic tokens from collapsing to one point in the following layers. Second, the semantic tokens can be forced to represent more local and distinguished features. Finally, the representation of semantic tokens is associated with the specific spatial locations, which is the basis to allow our method to be applied in local self-attention and downstream tasks. The initial cluster centers P then dynamically integrate the image tokens X according to semantic information by attention mechanism. In the first transformer layer, the processing of the generation of semantic tokens can be written as

$$\hat{S^1} = MHA(P, X, X) + P, \quad S^1 = FFN(\hat{S^1}) + \hat{S^1}, \tag{1}$$

where MHA and FFN are short for multi-head attention layer and feed-forward network, respectively, and the triplet input of MHA are queries, keys, and values in turn. All the norm layers are omitted in all the equations for brevity. The initial cluster centers are produced in a window by an adaptive spatial pooling layer or an intra and inter-window spatial pooling, while the semantic tokens are generated from a global receptive field by a dynamic attention layer to ensure that they can extract high-level semantic representation. In order to further strengthen the clustering effect, we use the second transformer layer to repeat the clustering operation and guide the semantic tokens to extract global information. In this transformer, the semantic tokens are updated as:

$$\begin{aligned} \hat{S^2} &= MHA(S^1 + G, Concat(S^1, X), Concat(S^1, X)) + S^1, \\ S^2 &= FFN(\hat{S^2}) + \hat{S^2}, \end{aligned} \tag{2}$$



(a) STViT (b) STViT-R

Figure 2. The architectures of our STViT and STViT-R.

where $G \in \mathbb{R}^{N_s \times C}$ is the global cluster centers initialized by Gaussian noise and $Concat(\cdot)$ is a concatenation operation. The global cluster centers G are responsible for global semantic information extraction like the class token. Although S^1 and G are summed together as the queries, they can be decoupled in the attention computation as:

$$A_s = (S^1 \cdot W_q) \cdot ((S^1 + X) \cdot W_k), \quad A_g = (G \cdot W_q) \cdot ((S^1 + X) \cdot W_k), \\ A = Softmax(A_s + A_g), \tag{3}$$

where W_q and W_k are the linear projection weights of queries and keys. We can see that S^1 and G integrate the keys to generate A_s and A_g independently and just share a Softmax operation to produce the final attention map A. Therefore, besides spatial semantic information, the semantic tokens also incorporate global semantic information with a negligible additional overhead. Though very similar, the global cluster centers are actually different from the learned positional encoding. We do not add the global cluster centers to keys. Moreover, it will be shown in the experiments that inserting the actual learned positional encoding will cause an accuracy drop. The number of transformer layers in STGM is flexible. More transformer layers can be associated for clustering. Two transformer layers are employed by default, *i.e.*, S^2 is the output of the STGM. Note that the image tokens are not updated in the STGM.

Intra and inter-window spatial pooling. Give $H \times W$ feature map X, we generate $N_s(w_s \times w_s)$ initial cluster centers. We uniformly split the X into $w_s \times w_s$ windows $[X_w^i]_{i=1}^{N_s}$ with size $\frac{H}{w_s} \times \frac{W}{w_s}$ and each window generates one initial cluster center. To represent abundant semantic information, we take into account intra and inter-window relations, i.e., integrating important tokens in the window and maximizing the distance among initial cluster centers in different windows. Specifically, we formulate an intra-window function f_{intra} to produce a mask, $M_i = f_{intra}(X_w^i)$, which projects the input window from $\mathbb{R}^{\frac{H}{w_s} \times \frac{W}{w_s} \times C} \rightarrow \mathbb{R}^{\frac{H}{w_s} \times \frac{W}{w_s}}$. The idea is to let the intra-window function f_{intra} adaptively select a combination of informative tokens in X_w^i, which is implemented by

$$M_i = Conv(GeLU(LayerNorm(DepthwiseConv(X_w^i)))). \tag{4}$$

Then, we compute each integrated token $\hat{P}_i = Softmax(M_i) \cdot X_w^i$ from each window, and arrange $[\hat{P}_i]_{i=1}^{N_s}$ by spatial structure to form the 2D tensor $\hat{P} \in \mathbb{R}^{w_s \times w_s \times C}$. We adopt an inter-window function f_{inter} to compute the inter-window relations and generate the offset, $O = f_{inter}(\hat{P})$,to revise the mask M. The implementation of f_{inter} is similar to Eq. 4, except the mapping input from $\mathbb{R}^{w_s \times w_s \times C} \rightarrow \mathbb{R}^{w_s \times w_s \times \frac{HW}{w_s^2}}$, where $\frac{HW}{w_s^2}$ is the number of tokens in each window. For each window, the corresponding $O_i \in \mathbb{R}^{\frac{HW}{w_s^2}}$ is sliced from O and reshaped to $\mathbb{R}^{\frac{H}{w_s} \times \frac{W}{w_s}}$. The O_i is used to revise M_i. The final initial cluster center P_i is computed by

$$P_i = Softmax(M_i + O_i) \cdot X_w. \tag{5}$$

Our f_{intra} and f_{intra} are super lightweight and the introducing parameters can be negligible. For example, on DeiT-T, the parameters only increase by 0.05%.

3.2. STViT in local vision transformers.

Local self-attention has been widely used in current ViT models to balance efficiency and accuracy. As the generation of semantic tokens in STViT is flexible in space, it can be naturally applied in local self-attention. Suppose each local self-attention layer contains N_w windows with size $w \times w$, we initialize $w_s \times w_s$ cluster centers in each window by our intra and inter-window spatial pooling. The total number of semantic tokens N_s would be $w_s \times w_s \times N_w$. w and w_s are set as 7 and 3 by default separately. As a result, our method compresses more than 80% image tokens in local self-attention. In the STGM, although initial cluster centers are from $w \times w$ windows, we use larger windows with size $w_k \times w_k$ to obtain keys and values in Eq. 1 and Eq. 2 to mitigate the effect of limited window size. Other operations in STGM are kept the same as Section 3.1.

In the local ViT models, each local transformer layer is normally followed by a cross-window connection layer, such as a shift window transformer layer following a local transformer layer on Swin Transformer [26]. In our method, the attention is computed within $w_s \times w_s$ window in the local self-attention layer, and the cross-window connection can be achieved by computing self-attention in a larger-size

($e.g.$, $4 \times w_s$) sliding window because of the rare number of tokens in each window. For the low-resolution input, our cross-window connection layer is equal to a global self-attention layer.

3.3. STViT for downstream tasks

Our method significantly reduces the computation cost by using a small number of semantic tokens, while its side effect is losing nearly all the detailed position information. High-level vision tasks, such as object detection and instance segmentation, are difficult to be executed on this extremely incomplete feature map. This issue also exists in previous works, which hinders the application of token sparsification methods. To solve this issue, we design a STViT-R network based on STViT to restore the original spatial resolution from the semantic tokens.

Our STViT-R shown in Figure 2b has two modifications compared with STViT. First, we adopt a recovery module to restore the spatial resolution from semantic tokens; second, we regroup the transformer layers and construct dumbbell units composed of our STViT-R.

Recovery module. In the recovery module, only the self-attention layer is employed to restore the spatial resolution without any additional networks. The image tokens X and semantic tokens S are partitioned as N_w^r windows of size $w^r \times w^r$ and $w_s^r \times w_s^r$, respectively. The image tokens in each window aggregate the semantic tokens in the corresponding window, which is represented as:

$$\hat{X} = MHA(X, S, S) + X, \quad X = FFN(\hat{X}) + \hat{X}. \quad (6)$$

This is a reverse operation of the generation of semantic tokens, using high-level semantic information to boost the image tokens.

Dumbbell unit. The transformer layers are regrouped into multiple dumbbell units in our STViT-R. Each dumbbell unit consists of four parts. The transformers in the first part are responsible for coping with image tokens; the second part is the semantic token generation module; the transformer layers in the third part deal with semantic tokens; the last part is the recovery module. Take the application on Swin-S (STViT-R-Swin-S) as an example. One, two, two and one transformer layers are allocated for these four parts, respectively. In total, each dumbbell unit is composed of 6 transformer layers. We concatenate three dumbbell units in Stage 3 of Swin-S. In each dumbbell unit, the intermediate transformer layers process semantic tokens with high-level semantic information to save computational cost, and the complete spatial resolution is recovered at the end. By repeating multiple dumbbell units, the detailed spatial information will be preserved by the network, which can not only enhance the classification but also serve downstream tasks.

4. Experiments

STViT will first be applied in two representative ViT models, DeiT [35] and Swin [26] for image classification and video recognition. To validate that our method is effective in downstream tasks, STViT-R is then performed on object detection and instance segmentation tasks.

4.1. Image classification

Settings. For image classification, all the models are trained on the ImageNet [13] with 1.28M training images and 50K validation images from 1,000 classes. By default, the semantic token generation module (STGM) employs the 5^{th} and 6^{th} transformer layers of DeiT (with 12 layers in total), employs the 3^{th} and 4^{th} transformer layers of Stage 3 of Swin-T (with 12 layers in total), and employs the 11^{th} and 12^{th} transformer layers of Stage 3 of Swin-S and Swin-B (with 24 layers in total). The image resolution in training and inference is 224×224 unless otherwise specified. The batch size is 1,024. All the models are trained from scratch for 300 epochs, and the augmentation and regularization strategies follow the original papers of DeiT and Swin. No knowledge distillation algorithms are used in our experiments. The classification is performed by applying a global average pooling layer on the output tokens of the last transformer layer, followed by a linear classifier. In evaluation, the top-1 accuracy using a single crop is reported. The FLOPs computations of this paper are measured by Fvcore[2]. Throughput is measured with the batch size of 128 on a V100 GPU.

On Swin [26], the semantic tokens are generated in Stage 3, and they are not downsampled in Stage 4 due to rare semantic tokens. The patch merging layer between Stage 3 and Stage 4 is replaced with a simple linear layer to double the number of channels. w_k is set as 10 and 14 for two transformer layers of STGM.

Results. One of the advantages of our method is that it can be applied to both global and local vision transformers to reduce computational complexity. Our main results on DeiT and Swin are summarized in Table 1 and Table 2, respectively. The results of LV-ViT [20] are illustrated in Supplementary A.3 due to limited space. We report the top-1 accuracy, FLOPs, and the throughput under different numbers of semantic tokens. On DeiT, the models with 16 semantic tokens achieve the same accuracy as the DeiT models with 196 tokens and save nearly 60% FLOPs on DeiT-T, DeiT-S, and DeiT-B. With more semantic tokens, the accuracy can consistently outperform the base models. For instance, STViT-DeiT-B with 36 semantic tokens surpasses DeiT-B by 0.4% accuracy with 52% FLOPs reduction.

The local vision transformer like Swin is already an efficient architecture compared to the global vision trans-

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

Model	Metrics	Base	No. of semantic tokens 16	36	64	100
STViT-DeiT-T	Top-1 Acc(%)	72.2	72.2(+0.0%)	72.7(+0.5)	73.0(+0.8)	73.2(+1.0)
	FLOPs(G)	1.26	0.53(-58%)	0.60(-52%)	0.71(-44%)	0.86(-32%)
	Throughput(img/s)	2752	5511(+101%)	4769(+74%)	4214(+53%)	3551(+29%)
STViT-DeiT-S	Top-1 Acc(%)	79.8	79.8(+0.0)	80.1(+0.3)	80.5(+0.7)	80.6(+0.8)
	FLOPs(G)	4.58	1.91(-58%)	2.20(-52%)	2.62(-43%)	3.16(-31%)
	Throughput(img/s)	1408	2891(+105%)	2542(+80%)	2229(+58%)	1837(+30%)
STViT-DeiT-B	Top-1 Acc(%)	81.8	81.8(+0.0)	82.2(+0.4)	82.6(+0.8)	82.7(+0.9)
	FLOPs(G)	17.58	7.31(-58%)	8.44(-52%)	10.04(-43%)	12.13(-31%)
	Throughput(img/s)	626	1308(+110%)	1150(+85%)	1087(+61%)	826(+33%)

Table 1. Applying STViT to DeiT-T, DeiT-S, and DeiT-B. The top-1 accuracy, complexity in FLOPs, and throughput are reported for different numbers of semantic tokens.

Model	Metrics	Base	Move STGM	No. of semantic tokens 4	9	16
STViT-Swin-T	Top-1 Acc(%)	81.3	81.0(-0.3%)	80.8(-0.5)	81.5(+0.2)	81.8(+0.5%)
	FLOPs(G)	4.5	3.14(-30%)	2.99(-34%)	3.43(-24%)	4.06(-10%)
	Throughput(img/s)	878	1124(+29%)	1128(+29%)	1061(+22%)	1008(+15%)
STViT-Swin-S	Top-1 Acc(%)	83.0	82.8(-0.2%)	82.4(-0.6%)	83.0(-0.0)	83.1(+0.1%)
	FLOPs(G)	8.7	5.95(-32%)	5.95(-32%)	6.53(-25%)	7.36(-15%)
	Throughput(img/s)	551	739(+35%)	732(+34%)	691(+26%)	657(+20%)
STViT-Swin-B	Top-1 Acc(%)	83.5	83.2(-0.3%)	83.0(-0.5)	83.4(-0.1)	83.7(+0.2%)
	FLOPs(G)	15.4	10.48(-32%)	10.48(-32%)	11.51(-25%)	12.97(-16%)
	Throughput(img/s)	415	558(+35%)	551(+33%)	521(+26%)	489(+19%)

Table 2. Applying STViT to Swin-T, Swin-S, and Swin-B. The top-1 accuracy, complexity in FLOPs, and throughput are reported for different numbers of semantic tokens in each window. *Base* indicates the corresponding original Swin model. *Move STGM* indicates changing the default position of STGM.

former, so the reduction of FLOPs on Swin models are smaller than on DeiT models. When 9 semantic tokens are used in each window, STViT-Swin models can reduce 25% FLOPs with negligible accuracy loss on all the model sizes. If the number of tokens is reduced to 4 in each window (16 in total), a significant accuracy drop will occur, which indicates that local vision transformers need more semantic tokens than global vision transformers. We can move the STGM towards shallow layers to attain a better complexity/accuracy trade-off. In Table 2, STGM is moved by one transformer layer on STViT-Swin-T and by two layers on STViT-Swin-S and STViT-Swin-B to save over 30% FLOPs with only about 0.3% accuracy drops (column of "Move STGM").

STViT-R equipped with recovery modules is designed for downstream tasks, while it also can perform image classification. The corresponding results are reported in Table 3. On both Swin-S and Swin-B, STViT-R can save 33% FLOPs with 0.3% accuracy drop. The hyper-parameters we used in STGM are as same as STViT-Swin.

These results demonstrate that our method achieves both effectiveness and efficiency by employing a few semantic tokens to replace original image tokens. Our method reveals that constructing the tokens with high-level semantic representation is more important than maintaining structured tokens in ViTs. As reflected by the throughput, our method does not have overhead of memory or deployment. Compared to the total parameters, the additional parameters introduced by intra and inter-window spatial pooling is negligible(less than 0.05%), so we do not show them.

Model	Top-1 Acc(%)	FLOPs(G)	Throughput
STViT-R-Swin-S	82.7(-0.3)	5.83(-33%)	717(+30%)
STViT-R-Swin-B	83.2(-0.3)	10.26(-33%)	539(+30%)

Table 3. STViT-R is evaluated on Swin-S and Swin-B on ImageNet. The top-1 accuracy, complexity in FLOPs, and throughput are reported.

Comparisons with existing token sparsification methods. In Table 4, we compare STViT with the state-of-the-art token sparsification methods on DeiT-S and DeiT-B. Due to different base models used by different methods, we

Model	Top-1 Acc	FLOPs(G)	△
DeiT-S			
DynamicViT [31]	79.3	2.9(-37%)	-0.5
IA-RED^2 [30]	79.1	3.2(-30%)	-0.7
PS-ViT [34]	79.4	2.6(-43%)	-0.4
TokenLearner* [32]	76.1	1.9(-44%)	-1.8
DGE* [33]	79.7	3.1 (-49%)	-0.6
A-ViT* [47]	78.6	3.6 (-39%)	-0.3
Evo-ViT [45]	79.4	3.0(-35%)	-0.4
EViT [24]	78.5	2.3(-50%)	-1.3
STViT(Ours)	79.8	1.91(**-58%**)	**-0.0**
DeiT-B			
IA-RED^2 [30]	80.3	11.8(-33%)	-1.5
DynamicViT [31]	81.3	11.2(-36%)	-0.5
PS-ViT [34]	81.5	9.8(-44%)	-0.3
TokenLearner* [32]	83.7	28.7(-48%)	-1.1
Evo-ViT [45]	81.3	10.2(-33%)	-0.5
EViT [24]	80.0	8.7(-51%)	-1.8
STViT(Ours)	81.8	7.31(**-58%**)	**-0.0**

Table 4. Comparisons with the state-of-the-art token sparsification methods on DeiT-S and DeiT-B. △ shows the accuracy difference between each model and its base model. $*$: their base models are not standard DeiT models.

adopt accuracy difference between each model and its base model Δ to evaluate them for fair. Results indicate that our method achieves the lowest accuracy drop Δ with the highest FLOPs reduction, outperforming all the state-of-the-art methods in both accuracy and efficiency significantly.

4.2. Video recognition

Setting. For video recognition, we apply our STViT to Video Swin [27]. All the models are pre-trained on ImageNet-1K and trained on Kinetics-400 [7]. We generate semantic tokens from each frame as illustrated in Section 3.2. The initialization from pre-trained models and other implementation details are as same as Video Swin [27].

Results. The results are presented in Table 6. On Swin-T and Swin-S, STViT-Swin can save about 27% FLOPs with 0.3% accuracy drop, which shows that our method works on video recognition.

4.3. Applications in object detection and instance segmentation

Settings. Experiments of object detection and instance segmentation are conducted on COCO 2017 [25]. We evaluate STViT-R with Swin in Cascade Mask R-CNN [4, 17] detection frameworks. The w_s is set to 3. The backbone models are pre-trained on ImageNet-1K and the pre-trained results are shown in Table 3. All the other settings follow Swin [26].

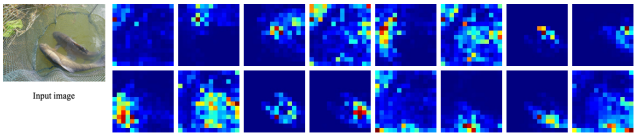


Figure 3. Visualization example of attention maps in the first attention layer of STGM.

Comparison to Swin Transformer. The performance of STViT-R-Swin using the Cascade Mask R-CNN framework with 3× schedule is shown in Table 5. Our method achieves better performance with more than 30% FLOPs reduction for backbone on bath object detection and instance segmentation. This validates that the recovery module and dumbbell unit can restore detailed spatial information, and the global context information integrated from the semantic tokens significantly benefits object detection. Ignoring spatial structure in the intermediate layers does not affect the object detection task, which is a meaningful fact to help design efficient object detection frameworks. Another interesting finding is that our method has a remarkable improvement for small object detection which is a challenging problem in the detection community. For instance, our STViT-R-Swin-S outperforms Swin-S by 1.5% on AP^b_s.

4.4. Ablation study

All the following ablation study of STViT and STViT-R are conducted on the DeiT-S and Swin-S, respectively.

Initialization analysis. The semantic tokens are the cluster centers recovered by attention layers. The initialization of cluster centers induces the representation of semantic tokens. Spatial and global initialization are adopted in Section 3.1 to guide the semantic tokens to integrate local and global semantic information separately. We compare different initialization components in Table 7. When performing single global initialization(3^{rd} row), we replace the spatial initial cluster centers with global initial cluster centers in the first transformer layer. The accuracy of using a single initialization method is far lower than using both, which shows the effectiveness of our initialization strategy. Our global initial cluster centers look similar to learned positional encoding since both use random initialization. To verify their distinction, we experiment with a real learned positional encoding by additionally adding the global initial cluster centers to keys, which causes 0.1% accuracy drop (the last row of Table 7). Therefore, our global initialization is different from learned positional encoding.

We visualize the attention maps of the first attention layer in the STGM in Figure 3. Because the queries of this layer are spatial initial cluster centers, these attention maps visualize the local semantic information integration by attention. The attention layer groups the semantic information according to the position of initial cluster centers, which ensures to extract fine-grained semantic information

	AP^b	AP^b_{50}	AP^b_{75}	AP^b_s	AP^m	AP^m_{50}	AP^m_{75}	AP^m_s	FLOPs(G)
Swin-S	51.8	70.4	56.3	35.2	44.7	67.9	48.5	28.8	194
STViT-R-Swin-S	51.8	70.6	56.1	36.7	44.7	67.8	48.6	29.0	134(-31%)
Swin-B	51.9	70.9	56.5	35.4	45.0	68.4	48.7	28.9	343
STViT-R-Swin-B	52.2	70.8	56.8	36.5	45.2	68.3	49.1	29.5	233(-32%)

Table 5. Results on COCO object detection and instance segmentation under Cascade Mask R-CNN with $3\times$ schedule. The FLOPs are measured for backbones.

Model	Top-1 Acc(%)	FLOPs(G)	Speed
Swin-T	78.8	88	779
STViT-Swin-T	78.5(-0.3)	64.4(-27%)	975(+25%)
Swin-S	80.6	166	456
STViT-Swin-S	80.3(-0.3)	120.5(-27%)	572(+25%)

Table 6. Applying STViT to Video Swin (Swin-T and Swin-S) on Kinetics-400. All the models are pre-trained on ImageNet-1K. The views are 4×3. The top-1 accuracy and complexity in FLOPs are reported. Speed is evaluated by FPS.

Spatial	Global	Learned	Top-1 Acc(%)
✓			79.4
	✓		78.7
✓	✓		79.8
✓		✓	79.7

Table 7. Accuracy with different initialization of STViT. *Spatial*, *Global*, and *Learned* indicate spatial initialization, global initialization, and learned positional encoding methods, respectively.

No. of transformers	2	3	4
Top-1 Acc(%)	79.8	79.5	79.6
FLOPs(G)	1.91	1.97	2.03

Table 8. Performance evaluation on different numbers of transformer layers in STGM. Keeping the base module containing four transformer layers unchanged.

	STViT-R	w/o DU	Reusing ST
AP^b	51.8	51.4	51.6
AP^m	44.7	44.4	44.5

Table 9. Ablation study on STViT-R w/o dumbbell units (*w/o DU*) and reusing semantic tokens (*Reusing ST*).

and keep the difference among semantic tokens. The attention map in the second attention layer is visualized in Supplementary A.5, which reveals that the network fixedly selects particle semantic tokens to represent global semantic information. We also show the semantic representation of image tokens in the same transformer layer on DeiT in Supplementary A.5. Compared to original image tokens, our semantic tokens in Figure 3 show more high-level semantic information.

Number of transformer layers in STGM. Two transformer layers are adopted in the STGM by default. The effects of employing different numbers of transformer layers are shown in Table 8. The extra layers are from the ones behind STGM to keep the total number of layers unchanged. More transformer layers do not bring improvement.

The effectiveness of the dumbbell unit. To verify the effectiveness of our dumbbell unit, we experiment STViT-R without dumbbell units, *i.e.*, STViT equipped with only the recovery module. We employ 6^{th} and 7^{th} transformer layers to construct STGM and the last transformer layers to construct the recovery module in Stage 3. The FLOPs is the same as the full-model STViT-R for fair comparison. The results on the COCO are reported in Table 9. Inferior results demonstrate the effectiveness of the dumbbell unit.

Reusing semantic tokens in dumbbell units. Semantic tokens are generated in each dumbbell unit. If they are produced only once in the first dumbbell unit and reused as initial cluster centers in the subsequent dumbbell units, the result is shown in Table 9 with a slight performance drop.

5. Conclusion

In this paper, we propose a simple and effective token sparsification method, semantic token vision transformer (STViT). Our method utilizes the clustering property of self-attention to generate a few semantic tokens with high-level information representation to replace the redundant image/video tokens, which can be applied in both global and local vision transformers. By simply configuring the recovery module, our method can be successfully applied to downstream tasks. Extensive experiments demonstrate that our method achieves better accuracy along with less inference time in most cases. The success in downstream tasks significantly boosts the development of token sparsification methods. We hope that this work can inspire more future research to pay much attention to high-level semantic representation in ViTs.

Acknowledgement

This project is supported by the National Research Foundation, Singapore under its NRFF Award NRF-NRFF13-2021-0008, and Mike Zheng Shou's Start-Up Grant from NUS. Computation was partially performed on resources of the National Supercomputing Centre, Singapore. Shuning was supported by Alibaba Research Intern Program.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Luvcić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 2

[2] Song Bai, Philip Torr, et al. Visual parser: Representing part-whole hierarchies with transformers. *arXiv preprint arXiv:2107.05790*, 2021. 3

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 2

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 7

[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. 2

[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. 2

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 7

[8] Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[9] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021. 2

[10] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 433–442, 2019. 3

[11] 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, 2021. 2

[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. 2

[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. 5

[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. *arXiv preprint arXiv:2107.00652*, 2021. 2

[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. 1, 2

[16] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Juergen Gall. Adaptive token sampling for efficient vision transformers. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 3

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 7

[18] 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. 2

[19] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021. 3

[20] 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. 2, 5

[21] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Bin Ren, Minghai Qin, Hao Tang, and Yanzhi Wang. Spvit: Enabling faster vision transformers via soft token pruning. *arXiv preprint arXiv:2112.13890*, 2021. 3

[22] 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. 2

[23] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9167–9176, 2019. 3

[24] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. 1, 2, 3, 7

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 7

[26] 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. 1, 2, 4, 5, 7

[27] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022. 7

[28] Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer. Luna: Linear unified nested attention. *Advances in Neural Information Processing Systems*, 34:2441–2453, 2021. 3

[29] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. *arXiv preprint arXiv:2111.15668*, 2021. 3

[30] Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. Ia-red ²: Interpretability-aware redundancy reduction for vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021. 3, 7

[31] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34, 2021. 3, 7

[32] Michael S. Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2, 3, 7

[33] Lin Song, Songyang Zhang, Songtao Liu, Zeming Li, Xuming He, Hongbin Sun, Jian Sun, and Nanning Zheng. Dynamic grained encoder for vision transformers. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 1, 3, 7

[34] 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. 3, 7

[35] 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. 1, 2, 5

[36] 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. 1, 2

[37] 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. 2

[38] Pichao Wang, Xue Wang, Hao Luo, Jingkai Zhou, Zhipeng Zhou, Fan Wang, Hao Li, and Rong Jin. Scaled relu matters for training vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 1

[39] 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. 2

[40] Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer hinging on cross-scale attention. *arXiv preprint arXiv:2108.00154*, 2021. 2

[41] Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[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. 2

[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. 1, 2

[44] 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. 2

[45] 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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 1, 2, 3, 7

[46] 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. 2

[47] Hongxu Yin, Arash Vahdat, Jose M 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*, pages 10809–10818, 2022. 3, 7

[48] Tan Yu, Gangming Zhao, Ping Li, and Yizhou Yu. Boat: Bilateral local attention vision transformer. *arXiv preprint arXiv:2201.13027*, 2022. 2

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

[50] David Junhao Zhang, Kunchang Li, Yali Wang, Yunpeng Chen, Shashwat Chandra, Yu Qiao, Luoqi Liu, and Mike Zheng Shou. Morphmlp: An efficient mlp-like backbone for spatial-temporal representation learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 230–248. Springer, 2022. 2

[51] Songyang Zhang, Xuming He, and Shipeng Yan. Latentgnn: Learning efficient non-local relations for visual recog-

nition. In *International Conference on Machine Learning*, pages 7374–7383. PMLR, 2019. 3

[52] 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. 2

[53] 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. 2

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

[55] 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. 2

[56] Zhuofan Zong, Kunchang Li, Guanglu Song, Yali Wang, Yu Qiao, Biao Leng, and Yu Liu. Self-slimmed vision transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pages 432–448. Springer, 2022. 1, 2, 3