EVIT: EXPEDITING VISION TRANSFORMERS VIA TOKEN REORGANIZATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision Transformers (ViTs) take all the image patches as tokens and construct multi-head self-attention (MHSA) among them. A complete leverage of these image tokens brings redundant computations since not all the tokens are attentive in MHSA. Examples include that tokens containing semantically meaningless or distractive image background do not positively contribute to the ViT model predictions. In this work, we propose to reorganize image tokens during the feed-forward process of ViT models. Our token reorganization method is integrated into ViT during training. For each forward inference, we identify attentive image tokens between the MHSA and FFN (i.e., feed-forward network) modules. The attentiveness identification of image tokens is guided by the corresponding class token. Then, we reorganize image tokens by preserving attentive image tokens and fusing inattentive ones to expedite subsequent MHSA and FFN computations. To this end, our method improves ViTs from two perspectives. First, under the same amount of input image tokens, our method reduces MHSA and FFN computation for efficient inference. For instance, the inference speed of DeiT-S is increased by 50% while its recognition accuracy is decreased by only 0.3% for ImageNet classification. Second, by maintaining the same computational cost, our method empowers ViTs to take more image tokens as input for recognition accuracy improvement, where the image tokens are from higher resolution images. An example is that we improve the recognition accuracy of DeiT-S by 1% for ImageNet classification at the same computational cost of a vanilla DeiT-S. Meanwhile, our method does not introduce more parameters to ViTs. Experiments on the standard benchmarks show the effectiveness of our method. Code will be made available.

1 INTRODUCTION

Computer vision research has evolved into Transformers since ViTs (Dosovitskiy et al., 2021). Equipped with global self-attention, ViTs have shown impressive capability upon local convolution (i.e., CNNs) on prevalent visual recognition scenarios, including image classification (Dosovitskiy et al., 2021; Touvron et al., 2021a; Jiang et al., 2021; Graham et al., 2021), object detection (Carion et al., 2020), and semantic segmentation (Xie et al., 2021; Liu et al., 2021; Wang et al., 2021a;c), with both supervised and unsupervised (self-supervised) training (Caron et al., 2021) configurations. Based on the main spirit of ViTs (i.e., MHSA), there are wide investigations (Liu et al., 2021; Yuan et al., 2021; Chu et al., 2021; Wang et al., 2021a; Xie et al., 2021; Han et al., 2021a) to explore network structure of ViT models for continuous recognition performance improvement.

Along with the development of ViT models, the computation burden is becoming an issue. The global self-attention between image tokens and long-range dependency make the model converge slow compared to CNNs. As illustrated in (Dosovitskiy et al., 2021), training a ViT from scratch typically requires larger datasets (e.g., ImageNet-21k and JFT-300M) than those of CNNs (e.g., CIFAR-10/100 and ImageNet-1k). Also, using more training iterations is a necessity for network convergence (Dosovitskiy et al., 2021; Touvron et al., 2021a). Without such large scale training, the ViT models are not fully exploited and perform inferior on visual recognition scenarios. These issues have made it necessary to expediting ViTs.

The model acceleration of ViTs is important to reduce computational complexity. However, there are few studies focused on ViT acceleration. This is because the significant model difference between

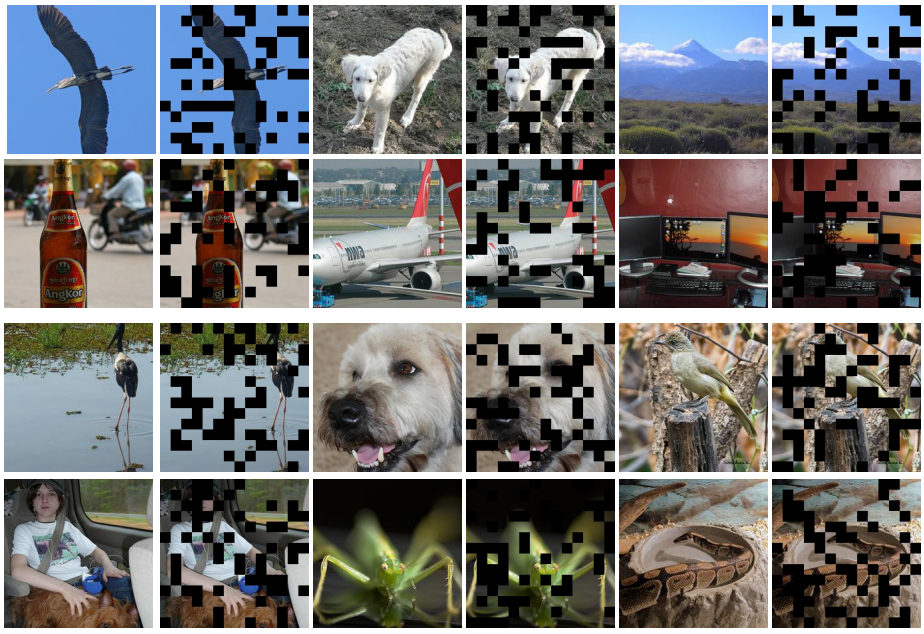

Figure 1: ViT predictions with incomplete input image tokens. The DeiT-S from (Touvron et al., 2021a) is used to perform the experiment. On the first two rows, removing image tokens unrelated to the visual content of the corresponding category does not deteriorate ViT predictions. On the last two rows, removing related image tokens makes ViT predict incorrectly.

CNNs and ViTs prevents CNN model acceleration (e.g., pruning and distillation) from applying on ViTs. Nevertheless, we analyze ViT from another perspective. We observe that not all image tokens in ViTs contribute positively to the final predictions. Figure 1 shows some examples where part of the input image tokens are randomly dropped out. On the last two rows, removing image tokens related to the visual content of the corresponding category makes the ViT predict incorrectly. In comparison, removing unrelated image tokens do not affect ViT predictions as shown on the first two rows. On the other hand, we notice that ViTs divide images into non-overlapping tokens and perform self-attention (Vaswani et al., 2017) computation on these tokens. A notable characteristic of self-attention is that it can process varying number of tokens. These observations motivate us to reorganize image tokens for ViT model accelerations.

In this work, we propose a token reorganization method to identify and fuse image tokens. Given all the image tokens as input, we compute token attentiveness between these tokens and the class token for identification. Then, we preserve the attentive image tokens and fuse the inattentive tokens into one token to allow the gradient back-propagate through the inattentive tokens for better attentive token identification. In this way, we gradually reduce the number of image tokens as the network goes deeper to decrease computation cost. Also, the capacity of the ViT backbone can be flexibly controlled via the identification process where no additional parameters are introduced. We adopt our token reorganization method on representative ViT models (i.e., DeiT (Touvron et al., 2021a) and LV-ViT (Jiang et al., 2021)) for ImageNet classification evaluation. The experimental results show our advantages. For instance, with the same amount of input image tokens, our method speeds up the DeiT-S model by 50%, while only sacrificing 0.3% recognition accuracy on ImageNet classification. On the other hand, we extend our methods to boost the ViT model recognition performance under the same computational cost. By increasing the input image resolution, our method facilitates Vision Transformers in taking more image tokens to achieve higher classification accuracy. Numerically, we improve the ImageNet classification accuracy of the DeiT-S model by 1% under the same computational cost. Moreover, by using an oracle ViT to guide the token reorganization process, our method can increase the accuracy of the original DeiT-S from 79.8% to 80.7% while reducing its computation cost by 36% under the multiply accumulate computation (MAC) metric.

2 RELATED WORK

2.1 VISION TRANSFORMERS

Transformers (Vaswani et al., 2017) have drawn much attention to computer vision recently due to its strong capability of modeling long-range relation. A few attempts have been made to add self-attention layers or Transformers on top of CNNs in image classification (Hu et al., 2019), object detection (Carion et al., 2020), segmentation (Wang et al., 2021c), image retrieval (Lu et al., 2019) and even video understanding (Sun et al., 2019; Girdhar et al., 2019). Vision Transformer (ViT) (Dosovitskiy et al., 2021) first introduced a set of pure Transformer backbones for image classification and its follow-ups modify the ViT architecture for not only better visual recognition (Touvron et al., 2021a; Yuan et al., 2021; Zhou et al., 2021) but many other high-level vision tasks, such as object detection (Carion et al., 2020; Zhu et al., 2020; Liu et al., 2021), semantic segmentation (Wang et al., 2021a; Xie et al., 2021; Chu et al., 2021), and video understanding (Bertasius et al., 2021; Fan et al., 2021). Vision Transformers have shown its strong potential as an alternative to the previously dominant CNNs.

2.2 MODEL ACCELERATION

Neural networks are typically overparameterized (Allen-Zhu et al., 2019), which results in significant redundancy in computation in deep learning models. To deploy the deep neural networks on mobile devices, we must reduce the storage and computational overhead of the networks. Many adaptive computation methods are explored (Bengio et al., 2015; 2013; Wang et al., 2018; Graves, 2016; Hu et al., 2020; Wang et al., 2020b; Han et al., 2021b) to alleviate the computation burden. Parameter pruning (Srinivas & Babu, 2015; Han et al., 2015; Chen et al., 2015b) reduces redundant parameters which are not sensitive to the final performance. Some other methods leverage knowledge distillation (Hinton et al., 2015; Romero et al., 2014; Luo et al., 2016; Chen et al., 2015a) to obtain a small and compact model with distilled knowledge of a larger one. These model acceleration strategies are limited to convolutional neural networks.

There are also some attempts to accelerate the computation of the Transformer model, including proposing more efficient attention mechanisms (Wang et al., 2020a; Kitaev et al., 2020; Choromanski et al., 2020) and the compressed Transformer structures (Liu et al., 2021; Heo et al., 2021; Wang et al., 2021a). These methods mainly focus on reducing the complexity of the network architecture through artificially designed modules. Another approach to ViT acceleration is reducing the number of tokens involved in the inference of ViTs. Notably, Wang et al. (2021b) proposed a method to dynamically determine the number of patches to divide on an image. The ViT will stop inference for an input image if it has sufficient confidence on the prediction of the intermediate outputs. Another related work is DynamicViT (Rao et al., 2021), which introduces a method to reduce token for a fully trained ViT, where an extra learnable neural network is added to ViT to select a subset of tokens. Our work provides a novel perspective for reducing the computational overhead of inference by proposing a token reorganization method to progressively reduce and reorganize image tokens. Unlike DynamicViT, our method does not need a fully trained ViT to help the training and brings no additional parameters into ViT.

3 TOKEN REORGANIZATIONS

Our method EViT is built upon ViT (Dosovitskiy et al., 2021) and its variants for visual recognition. We first review ViT and then present how to incorporate our method into the ViT training procedure. Each component of EViT, including the attentive token identification and inattentive token fusion, will be elaborated. Furthermore, we analyze the effectiveness of our method by visualizing the attentive tokens at different layers and discuss training on higher resolution images with EViT.

3.1 VIT OVERVIEW

Vision Transformers (ViTs) are first introduced by Dosovitskiy et al. (2021) into visual recognition. They perform tokenization by dividing an input image into patches and projecting each patch to a token embedding. An extra class token `[CLS]` is added to the set of image tokens and is respon-

Token keep rate	1.0	0.9	0.8	0.7	0.6	0.5
Top-1 ACC (%)	79.8	79.7(-0.1)	79.2(-0.6)	78.5(-1.3)	76.8(-3.0)	73.8(-6.0)

Table 1: ImageNet classification accuracy of a straightforward inattentive token removal for a trained DeiT-S (Touvron et al., 2021a). The inattentive tokens are directly removed based on the attention from the class token to other tokens at the 4^{th}, 7^{th} and 10^{th} layers.

sible for aggregating global image information and final classification. All of the tokens are added by a learnable vector (i.e., positional encoding) and fed into the sequentially-stacked Transformer encoders consisting of a multi-head self-attention (MHSA) layer and a feed-forward network (FFN). In MHSA, the tokens are linearly mapped and further packed into three matrices, namely $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$. The attention operation is conducted as follows.

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}})\boldsymbol{V}. \tag{1}$$

where d is the length of the query vector. The result of $\text{Softmax}(\boldsymbol{Q}\boldsymbol{K}^\top/\sqrt{d})$ is a square matrix which is called the attention map. The first row of attention map represents the attention from `[CLS]` to all tokens and will be used to determine the attentiveness (importance) of each token (detailed in the next subsection). The output tokens of MHSA are sent to FFN, consisting of two fully connected layers with a GELU activation layer (Hendrycks & Gimpel, 2016) in between. At the final Transformer encoder layer, the `[CLS]` token is extracted and utilized for object category prediction. More details of Transformers can be found in Vaswani et al. (2017).

3.2 Attentive token identification

Let n denote the number of input tokens to a ViT encoder. In the last encoder of ViT, the `[CLS]` token is taken out for classification. The interactions between `[CLS]` and other tokens are performed via the attention mechanism (Vaswani et al., 2017) in the ViT encoders:

$$\boldsymbol{x}_{\text{class}} = \text{Softmax}(\frac{\boldsymbol{q}_{\text{class}} \cdot \boldsymbol{K}^\top}{\sqrt{d}})\boldsymbol{V} = \boldsymbol{a} \cdot \boldsymbol{V}. \tag{2}$$

where $\boldsymbol{q}_{\text{class}}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ denote the query vector of `[CLS]`, the key matrix, and the value matrix, respectively, in an attention head. In other words, the output of the `[CLS]` token $\boldsymbol{x}_{\text{class}}$ is a linear combination of the value vectors $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n]^\top$, with the combination coefficients (denoted by $\boldsymbol{a}$ in Eq. 2) being the attention values from `[CLS]` with respect to all tokens. Since $\boldsymbol{v}_i$ comes from the i-th token, the attention value a_i (i.e., the i-th entry in $\boldsymbol{a}$) determines how much information of the i-th token is fused into the output of `[CLS]` (i.e., $\boldsymbol{x}_{\text{class}}$) through the linear combination. It is thus natural to assume that the attention value a_i indicate the importance of the i-th token.

Moreover, Caron et al. (2021) also showed that the `[CLS]` token in ViTs pays more attention (i.e., having a larger attention value) to class-specific tokens than the tokens on the non-object regions. To this end, we propose to use the attentiveness of the `[CLS]` token with respect to other tokens to identify the most important tokens. Based on these arguments, a simple method to reduce computation in ViT is to remove the tokens with the smallest attention values. However, we find that directly removing those tokens severely deteriorate the classification accuracy, as shown in Table 1. Therefore, we propose to incorporate image token reorganization during the ViT training process.

In multi-head self-attention layer, there are multiple heads performing the computation of Eq. 1 in parallel. Thus, there are multiple `[CLS]` attention vectors $\boldsymbol{a}^{(h)}$, $h = [1, \ldots, H]$, with H being the total number of attention heads (Vaswani et al., 2017). We compute the average attentiveness value of all heads by $\bar{\boldsymbol{a}} = \sum_{h=1}^{H} \boldsymbol{a}^{(h)}/H$. As shown in Figure 2, we identify and preserve the tokens corresponding to the k largest (top-k) elements in $\bar{\boldsymbol{a}}$ (k is a hyperparameter), which we call the attentive tokens, and further fuse the other tokens (which we call the inattentive tokens) into a new token. The fusion of tokens are detailed in the following paragraph. We define the token keeping rate as $\kappa = k/n$.

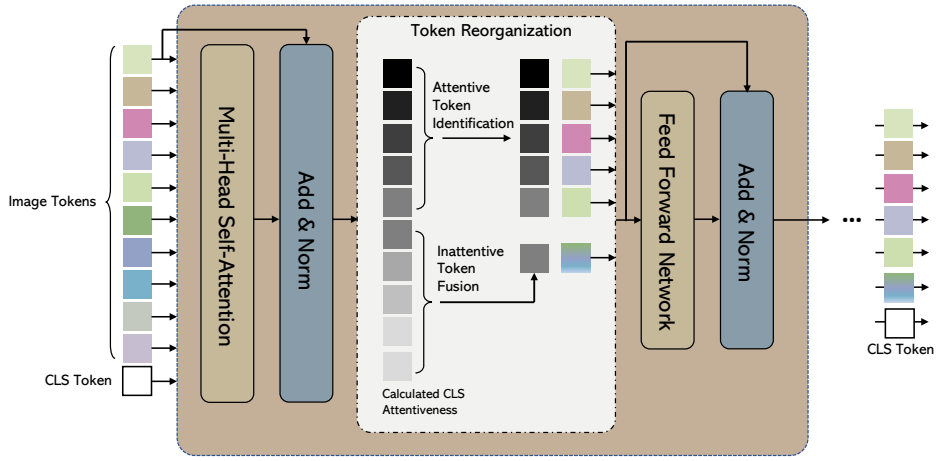

Figure 2: Token reorganization within a single Transformer encoder. Based on ViT (Dosovitskiy et al., 2021), we reorganize tokens in the original Transformer encoder. Specifically, we calculate the attentiveness of the class token with respect to each image token. Then, we use the attentiveness value as a criterion to identify the top-k attentive tokens and fuse the inattentive tokens.

3.3 Inattentive Token Fusion

At the initial training stage, the ViT model is not stable and attentive tokens are incorrectly identified. To mitigate this effect, we fuse the remaining tokens at the current stage to supplement attentive ones, as illustrated in Figure 2. The inattentive token fusion benefits our method to accurately identify attentive tokens. Meanwhile, the ViT training converges more efficiently.

We denote the indices set of the inattentive tokens as $\mathcal{N}$. The proposed inattentive token fusion is a weighted average operation, which can be written as follows:

$$\boldsymbol{x}_{\text{fused}} = \sum_{i \in \mathcal{N}} a_i \boldsymbol{x}_i \tag{3}$$

The fused token $\boldsymbol{x}_{\text{fused}}$ is appended to the attentive tokens and sent to the subsequent layers. The computation cost of token fusion is negligible compared to the bulk computation of ViT.

3.4 Analysis

Training with higher resolution images. Since our approach is efficient in processing image tokens, we are able to feed more tokens into an EViT while maintaining the computational cost at the same level of a vanilla ViT. A straightforward method to get more tokens is resizing the input images to a higher resolution and keeping the patch size unchanged. Note that these higher resolution images do not need to come from raw images with a larger resolution. In our vision recognition experiments on ImageNet, we simply resize the standard input images of size 224×224 to a larger spatial size (e.g., 256×256) via bicubic interpolation to obtain the higher resolution images, which are further divided into more tokens. Therefore, compared to a vanilla ViT, EViT uses *no* additional information for the images to obtain the prediction results in both training and inference. The experimental results in Table 6 validate the effectiveness of our proposed method.

Visualization. Our proposed EViT accelerates ViTs by identifying the attentive tokens and discarding the redundant calculation on inattentive image tokens. To further investigate the interpretability of EViT, we visualize the attentive token identification procedure in Figure 3. We present the original images and the attentive token identification results at different layers (e.g., the 4^{th}, 7^{th} and 10^{th} layer). It can be seen that as the network deepens, the inattentive tokens are gradually removed or fused, while the most informative/attentive tokens are identified and preserved. In this way, our proposed method facilitates the ViTs in focusing on class-specific tokens in images. The visualization results also validate that our EViT is effective in dealing with images with either simple backgrounds or complex backgrounds.

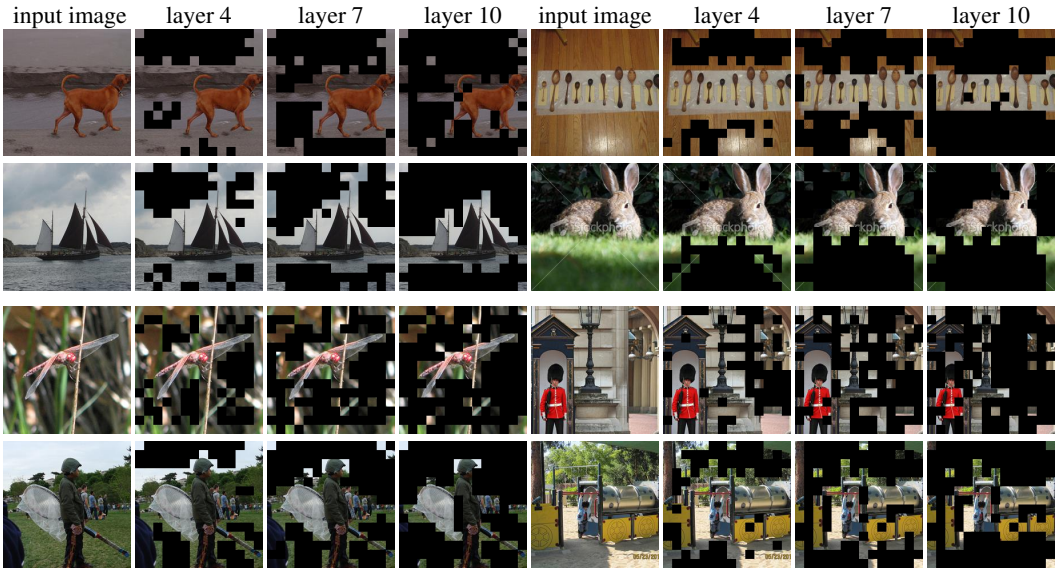

Figure 3: Visualization of inattentive tokens on EViT-DeiT-S with 12 layers. The masked regions represent the inattentive tokens that are fused into a new token. Our method can effectively identify inattentive tokens in images with either simple inattentive tokens (in the first two rows) or complex inattentive tokens (in the last two rows).

Table 2: Comparison on the two variants of EViT on DeiT-S (Touvron et al., 2021a). The results are the average of three independent trials for reducing the influence of randomness.

(a) **Attentive token identification w/o inattentive token fusion**

Keep rate	Top-1 ACC (%)	Top-5 ACC (%)	Throughput (images/s)	MACs
DeiT-S	79.8	94.9	2923	5.0
0.9	79.9 (+0.1)	94.9 (-0.0)	3201 (+10%)	4.3 (-14%)
0.8	79.7 (-0.1)	94.8 (-0.1)	3772 (+27%)	3.7 (-26%)
0.7	79.4 (-0.4)	94.7 (-0.2)	4249 (+45%)	3.2 (-36%)
0.6	78.9 (-0.9)	94.5 (-0.4)	4967 (+70%)	2.8 (-44%)
0.5	78.4 (-1.4)	94.1 (-0.8)	5325 (+82%)	2.5 (-50%)

(b) **Attentive token identification w/ inattentive token fusion**

Keep rate	Top-1 ACC (%)	Top-5 ACC (%)	Throughput (images/s)	MACs (G)
DeiT-S	79.8	94.9	2923	5.0
0.9	79.8 (-0.0)	95.0 (+0.1)	3197 (+9%)	4.3 (-14%)
0.8	79.8 (-0.0)	94.9 (-0.0)	3619 (+24%)	3.7 (-26%)
0.7	79.5 (-0.3)	94.8 (-0.1)	4385 (+50%)	3.2 (-36%)
0.6	78.9 (-0.9)	94.5 (-0.4)	4722 (+62%)	2.8 (-44%)
0.5	78.5 (-1.3)	94.2 (-0.7)	5408 (+85%)	2.5 (-50%)

4 EXPERIMENTS

Implementation details. We train all of the models on the ImageNet (Deng et al., 2009) training set with approximately 1.2 million images and report the accuracy on the 50k images in the test set. By default, the token identification module is incorporated into the 4^{th}, 7^{th} and 10^{th} layer of DeiT-S and Deit-B (with 12 layers in total) and incorporated into the 5^{th}, 9^{th} and 13^{th} layer of LV-ViT-S (with 16 layers in total). The image resolution in training and testing is 224×224 unless otherwise specified. For the training strategies and optimization methods, we simply follow those in the original papers of DeiT (Touvron et al., 2021a) and LV-ViT (Jiang et al., 2021). Since our method can be easily incorporated into these models without making substantial modification to them, the original training strategies work well with our method. Besides, we adopt a warmup strategy for the attentive token identification. Specifically, the keep rate of attentive tokens is gradually reduced from 1 to the target value with a cosine schedule. Unlike DynamicViT (Rao et al., 2021), we do not use a pretrained ViT to initialize our models in most experiments, except in the experiments with an oracle ViT (see the following paragraphs). We train the models with EViT from scratch for 300 epochs on 2 NVIDIA A100 GPUs and measure the throughput of the models on a single A100 GPU with a batch size of 128 unless otherwise specified.

Table 3: Results of EViT on DeiT-B (Touvron et al., 2021a) and LV-ViT-S (Jiang et al., 2021).

(a) Results of EViT on DeiT-B

Keep rate	**Top-1 ACC (%)**	**Top-5 ACC (%)**	**Throughput (images/s)**	**MACs (G)**
DeiT-B	81.8	95.6	1295	18.3
0.9	81.8 (-0.0)	95.6 (-0.0)	1441 (+11%)	16.0 (-12%)
0.8	81.7 (-0.1)	95.4 (-0.2)	1637 (+26%)	13.8 (-25%)
0.7	81.3 (-0.5)	95.3 (-0.3)	2053 (+59%)	12.0 (-34%)
0.6	80.9 (-0.9)	95.1 (-0.5)	2177 (+68%)	10.5 (-43%)
0.5	80.0 (-1.8)	94.5 (-1.1)	2482 (+92%)	9.2 (-50%)

(b) Results of EViT on LV-ViT-S

Keep rate	**Top-1 ACC (%)**	**Top-5 ACC (%)**	**Throughput (images/s)**	**MACs (G)**
LV-ViT-S	83.3	–	2112	6.6
0.7	83.0 (-0.3)	96.3	2954 (+40%)	4.7 (-29%)
0.5	82.5 (-0.8)	96.2	3603 (+71%)	3.9 (-41%)

Table 4: The performance of EViT-DeiT-S with different combinations of keep rates and reorganization positions (layers), which have the same level of inference throughput and MACs.

Reorganization layers	Keep rates	Top-1	Top-5	Throughput (img/s)	MACs (G)
[4, 7, 10]	[0.70, 0.70, 0.70]	79.50	94.77	4385	3.25
[5, 7, 10]	[0.64, 0.70, 0.70]	79.57	94.80	4271	3.23
[3, 7, 10]	[0.74, 0.70, 0.70]	79.47	94.72	4261	3.23
[4, 8, 10]	[0.70, 0.64, 0.70]	79.64	94.86	4299	3.25
[4, 6, 10]	[0.70, 0.75, 0.70]	79.54	94.78	4250	3.25

We report the main result of EViT on Tables 2 and 3. On both DeiT and LV-ViT, the proposed EViT achieves significant speedup while restricting the accuracy drop in a relatively small range. For example, DeiT-S trained with EViT with a keep rate of 0.7 increase the inference throughput by 50% while maintaining the Top-1 accuracy reduction within 0.3% on ImageNet. We plot these results in Figure 4, which shows EViT is very competitive against many vision models in terms of computation-accuracy trade-off.

Inattentive token fusion. As we have discussed in the previous section, random factors in ViTs training (such as random initialization) may interfere with the token selection process. To mitigate this problem, we propose inattentive token fusion, which fuse the non-topk tokens according to the attentiveness from the `[CLS]` token. We experimentally compare the proposed token reorganization method with and without inattentive token fusion. As shown in Table 2, token reorganization with inattentive token fusion generally outperforms the vanilla counterpart without inattentive token fusion, while the two has basically the same computational complexity and inference throughput. Moreover, we observe in our experiments that the vanilla token reorganization has a higher fluctuation in accuracy. On average, token reorganization method w/ inattentive token fusion has an standard deviation of 0.12 in accuracy, which is smaller than that w/o inattentive token fusion (0.15).

Token reorganization locations. It is possible that different combinations of keep rates and token reorganization layers can reach the same computational efficiency. For example, wee can a) move the reorganization operation one layer ahead and increase the keep rate, or b) move the reorganization operation one layer behind and decrease the keep rate, to keep the computation cost (approximately) unchanged. We train DeiT-S with different reorganization locations, each of which has a similar computational cost as the first one. To reduce the influence of randomness, we repeat the experiments and report the average over two trials in Table 4. The results show that moving reorganization operations to deeper layers slightly improve the accuracy over the standard configuration (the first row) used in our experiments, suggesting the possibility of further improvement of the proposed method. For simplicity, we did not search for better configurations and stick with the standard one.

Epochs of training. Since ViTs do not have inductive bias such as translational invariance processed by CNNs, they typically require more training data and/or training epochs to reach a comparable generalization performance as CNNs (Dosovitskiy et al., 2021; Touvron et al., 2021a). We find that training longer epochs continues to benefit ViTs in efficient computation regime. We train the DeiT-S model with 0.7 keep rate for longer epochs of 450 and 600, respectively. As shown in Table 5, their performance steadily improves with training epochs.

Table 5: Results of training EViT-DeiT-S with a keep rate of 0.7 for different epochs.

Keep rate	Epochs	Top-1	Top-5	MACs (G)
0.7	300	79.5	94.8	3.25
0.7	450	80.2	95.1	3.25
0.7	600	81.0	95.3	3.25

Table 6: Results of training/finetuning on high resolution images. EViT-DeiT-S and EViT-LV-ViT-S have a throughput/MACs comparable to the baselines while achieving better recognition accuracy on ImageNet. The number behind ↑ indicates the image size for finetuning for 100 epochs.

(a) DeiT-S

Model	Keep rate	Image size	Top-1 (%)	Top-5 (%)	img/s	MACs (G)
DeiT-S	1.0	224	79.8	94.9	2923	4.99
EViT	0.5	256	79.3	94.7	3788	3.33
EViT	0.5	288	80.1	95.0	3138	4.38
EViT	0.5	304	81.0	95.6	2905	4.97
EViT	0.6	256	80.0	95.0	3524	3.79
EViT	0.6	288	81.0	95.4	2927	4.99
EViT	0.7	272	80.3	95.3	2870	5.01

(b) LV-ViT-S

Model	Keep rate	Image size	Top-1 (%)	Top-5 (%)	img/s	MACs (G)
LV-ViT-S	1.0	224	83.3	–	2112	6.60
LV-ViT-S	1.0	224 ↑ 384	84.4	–	557	21.9
EViT	0.9	240	83.6	96.5	1956	6.79
EViT	0.8	256	83.6	96.6	1901	6.93
EViT	0.7	256	83.5	96.5	2102	6.22
EViT	0.7	272	83.7	96.6	1829	7.07
EViT	0.5	304	83.4	96.5	1758	7.38
EViT	0.7	272 ↑ 448	84.7	97.1	548	21.5

Training/Finetuning on higher resolution images. By fusing the inattentive tokens, We are able to feed EViT with more tokens under the same computational cost. Therefore, we train and/or finetune EViT on resized images with higher resolutions than the standard resolution of 224^2, and we report the results on Table 6. We can see that EViT in most tested cases performs favourably against a vanilla DeiT/LV-ViT, while having a comparable or higher inference throughput than the baselines. Notably, training EViT-LV-ViT-S on images of resolution 224^2 and further finetuning it on a higher resolution of 448^2 for another 100 epochs gives a very competitive Vision Transformer model, achieving an ImageNet top-1 accuracy of 84.7%, which is 0.3% higher than LV-ViT-S@384 with basically the same throughput and number of parameters. The experimental results validate our hypothesis that images typically contain tokens that are less informative and contribute little to the recognition task. Since ViTs perform global self-attention among all tokens in as early as the first layer, the early interaction and information exchange between tokens makes it possible to discard/fuse some of the least informative tokens in intermediate ViT layers since they have been "seen" by other tokens (including the `[CLS]` token). In comparison, the receptive field of a CNN at the shallow layers is relatively small due to the locality property of convolution, which makes it difficult to reduce computation at early stages.

Training with an oracle ViT. In EViT, the criterion of selecting tokens is the attention between the `[CLS]` token and other tokens. Therefore, it would be very helpful if we know the importance of each token to the prediction tasks in advance. To this end , we introduce an oracle ViT to guide

Table 7: Results of training EViT-DeiT-S and EViT-DeiT-B using DeiT-S as an oracle. Training for longer epochs continues to benefits the EViT in efficiency regime.

Model	Keep rate	Epochs	Top-1	Top-5	Throughput (img/s)	MACs (G)
DeiT-S	1.0	300	79.8	94.9	2923	4.99
EViT-DeiT-S w/o Oracle	0.7	300	79.5	94.8	4385	3.25
EViT-DeiT-S w/ Oracle	0.7	300	80.8	95.4	4385	3.25
EViT-DeiT-S w/ Oracle	0.7	450	81.0	95.5	4385	3.25
EViT-DeiT-S w/ Oracle	0.7	600	81.3	95.5	4385	3.25
DeiT-B	1.0	300	81.8	95.6	1295	18.3
EViT-DeiT-B w/o Oracle	0.7	300	81.3	95.3	2053	12.0
EViT-DeiT-B w/ Oracle	0.7	300	82.1	95.6	2053	12.0

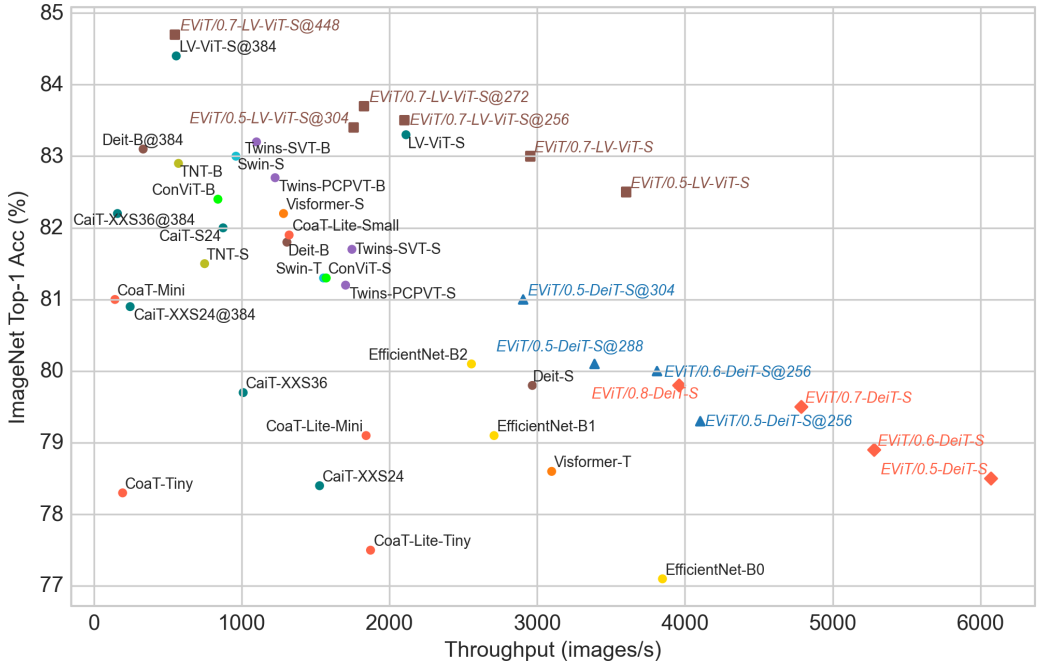

Figure 4: Comparison of different models with various accuracy-throughput trade-off. The proposed method EViT achieves better trade-off than the other methods (marked with circles). The throughput is measured on an NVIDIA A100 GPU using the largest possible batch size for each model. The input image size is 224^2 unless specified after the @. The comparing methods include DeiT (Touvron et al., 2021a), CaiT (Touvron et al., 2021b), LV-ViT (Jiang et al., 2021), CoaT (Xu et al., 2021), Swin (Liu et al., 2021), Twins (Chu et al., 2021), Visformer (Chen et al., 2021), ConViT (d'Ascoli et al., 2021), TNT (Han et al., 2021a), and EfficientNet (Tan & Le, 2019).

the `[CLS]` through the token selection process. A good oracle knows which tokens are important and which are not. For this purpose, we use a fully trained DeiT-S/B as an oracle and initialize EViT-DeiT-S/B with the parameters of the oracle ViT, such that the EViT know the which tokens contribute more to prediction result. We train EViT equipped with an oracle using the same training setup as training a vanilla EViT. As shown in Table 7, training with an oracle significantly improves the recognition accuracy of EViT.

5 Conclusion

In this paper, we present a token reorganization method. By identifying the tokens with highest attention from the class token, the proposed EViT reach a better trade-off between accuracy and efficiency than various Vision Transformer models. Moreover, we propose inattentive token fusion, which fuse the information from less informative tokens to a new token. Inattentive token fusion improves both the recognition accuracy and training stability. Experimentally, we apply the proposed token reorganization method to two variants of Vision Transformer, namely, DeiT (Touvron et al., 2021a) and LV-ViT (Jiang et al., 2021). In both variants, EViT achieves a significant speedup in inference while the reduction in recognition accuracy is relatively small. Moreover, when training on higher resolution images, EViT improves the accuracy to various extents while maintaining a similar or smaller computation cost as the original DeiT/LV-ViT models. Besides, when equipped with an oracle ViT which knows which tokens are more important, EViT can achieve further improvement on the trade-off of accuracy and efficiency. The proposed EViT can be easily adapted in ViTs and brings no additional parameters, nor does it require sophisticated training strategies. The proposed token reorganization method can serve as an effective acceleration approach for Vision Transformers.

REFERENCES

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 2019.

Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021.

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

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. *arXiv preprint arXiv:1511.05641*, 2015a.

Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *International conference on machine learning*, pp. 2285–2294, 2015b.

Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer. *arXiv preprint arXiv:2104.12533*, 2021.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

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. *Arxiv preprint 2104.13840*, 2021.

Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE CVPR*, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv:2104.11227*, 2021.

Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *IEEE CVPR*, 2019.

Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. *arXiv preprint arXiv:2104.01136*, 2021.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021a.

Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*, 2015.

Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *arXiv preprint arXiv:2102.04906*, 2021b.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *IEEE ICCV*, 2019.

Ting-Kuei Hu, Tianlong Chen, Haotao Wang, and Zhangyang Wang. Triple wins: Boosting accuracy, robustness and efficiency together by enabling input-adaptive inference. In *International Conference on Learning Representations*, 2020.

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. *arXiv preprint arXiv:2104.10858*, 2021.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

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. *arXiv preprint arXiv:2103.14030*, 2021.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

Ping Luo, Zhenyao Zhu, Ziwei Liu, Xiaogang Wang, and Xiaoou Tang. Face model compression by distilling knowledge from neurons. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

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.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Suraj Srinivas and R Venkatesh Babu. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*, 2015.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *IEEE ICCV*, 2019.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021a.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021b.

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

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020a.

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 *IEEE ICCV*, 2021a.

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 409–424, 2018.

Yue Wang, Jianghao Shen, Ting-Kuei Hu, Pengfei Xu, Tan Nguyen, Richard Baraniuk, Zhangyang Wang, and Yingyan Lin. Dual dynamic inference: Enabling more efficient, adaptive, and controllable deep inference. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):623–633, 2020b.

Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic vision transformers with adaptive sequence length. *arXiv preprint arXiv:2105.15075*, 2021b.

Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *IEEE CVPR*, 2021c.

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

Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. *arXiv preprint arXiv:2104.06399*, 2021.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.

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

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

A VISUALIZATION

In this part, we present more visualization results in Figure 5 to show the attentive token identification. The input images are randomly selected from ImageNet dataset. The results validate that our EViT is able to deal with different images from various categories.

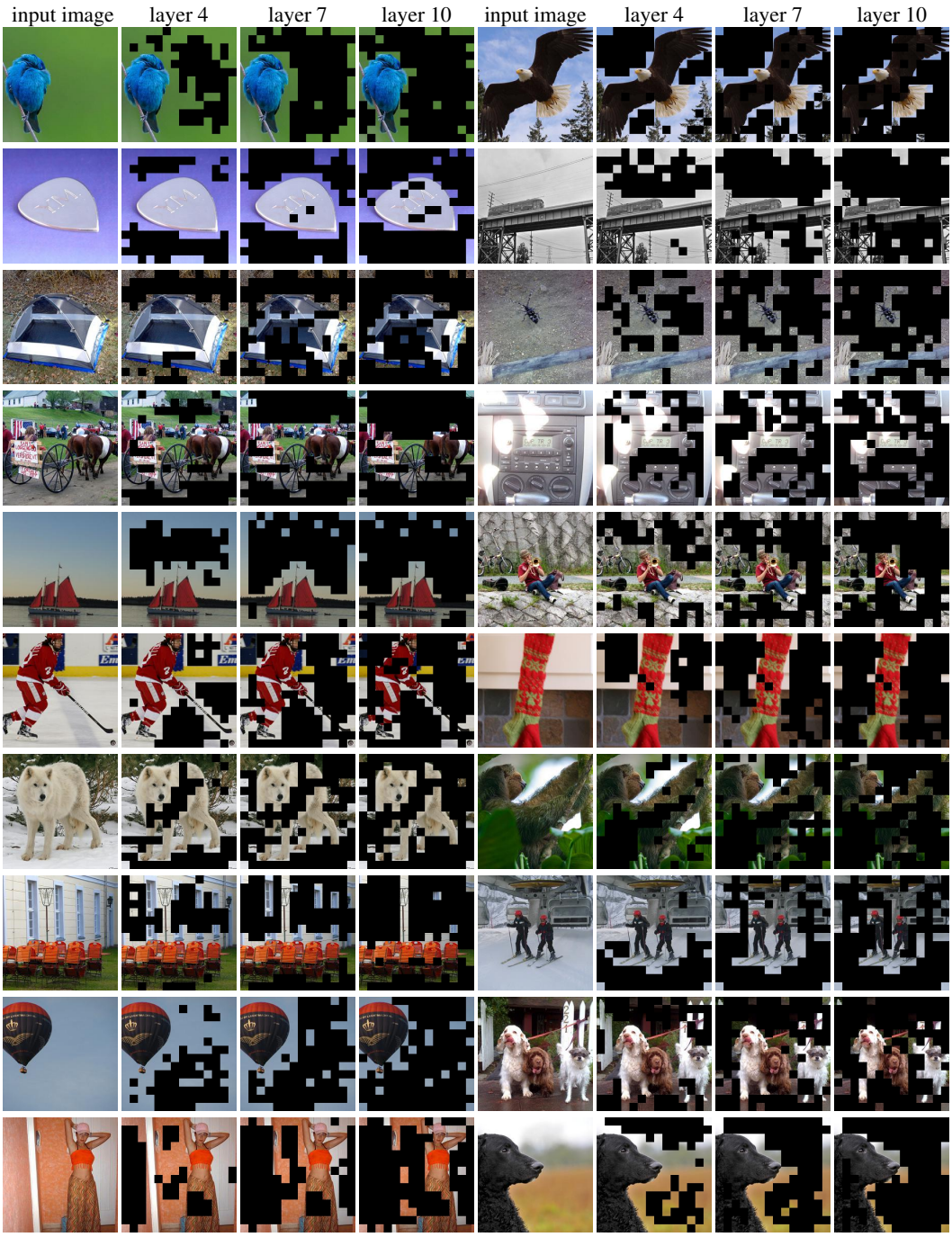

Figure 5: Extended visualization results of inattentive tokens on EViT-DeiT-S with 12 layers.. The regions without masks represent the attentive tokens. The masked regions denote the inattentive tokens that are fused into a new token. Our EViT is effective in dealing with images from different categories.

Algorithm 1: PyTorch-like pseudocode of EViT for a ViT encoder.

```
# H: number of attention heads
# N: number of input tokens
# C: the dimension of token vector
# k: the token keep rate
# x: the input tokens with shape [N, C], with the first being the [CLS]
    token
# fc_q, fc_k, fc_v: linear transforms for query, key, and value of self-
    attention
# @: matrix multiplication
# proj: linear projection in self-attention
# norm: layer normalization
# ffn: feed-forward network

# initialize
avg_cls_attn = zeros(N-1)
x_out = []
x_residual = x
x = norm(x)

## multi-head self-attention computation
for i in range(0, H): # compute self-attention for each attention head
   # linearly map the tokens to query, key, and value matrices
   q, k, v = fc_q[i](x), fc_k[i](x), fc_v[i](x)
   # compute the attention map
   attn = (q @ k.transpose()) / sqrt(C/H)
   attn = softmax(attn, dim=1)

   x_head = attn @ v
   x_out.append(x_head)

   # compute the [CLS] attentiveness w.r.t. other tokens,
   # without using [CLS] attention to itself
   cls_attn = attn[0, 1:]
   avg_cls_attn += cls_attn

# concatenate the output tokens of all heads
x = concat(x_out, dim=1)

x = proj(x) # shape: [N, C]
x = x + x_residual

# average the [CLS] attentiveness over all heads
avg_cls_attn /= H

# sort the avg_cls_attn in descending order
sorted_cls_attn, idx = sort(avg_cls_attn)

# compute the number of attentive tokens, without counting the [CLS] token
K = ceil(k * (N - 1))

topk_attn, topk_idx = sorted_cls_attn[:K], idx[:K]
non_topk_attn, non_topk_idx = sorted_cls_attn[K:], idx[K:]

# separate [CLS] token and other tokens
cls_token = x[0:1]
x_without_cls = x[1:]

# obtain the attentive and inattentive tokens
attentive_tokens = x_without_cls[topk_idx]
inattentive_tokens = x_without_cls[non_topk_idx]

# compute the weighted combination of inattentive tokens
fused_token = non_topk_attn @ inattentive_tokens

# concatenate these tokens
x_new = concat([cls_token, attentive_tokens, fused_token], dim=0)

x_residual = x_new
x_new = norm(x_new)
x_new = ffn(x_new)
x_new = x_new + x_residual
return x_new
```