Predicting Token Impact Towards Efficient Vision Transformer

Hong Wang[1] Su Yang[1] * Xiaoke Huang[1] Weishan Zhang[2]
[1] Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University
[2] School of Computer Science and Technology, China University of Petroleum
hongwang21@m.fudan.edu.cn

Abstract

Token filtering to reduce irrelevant tokens prior to self-attention is a straightforward way to enable efficient vision Transformer. This is the first work to view token filtering from a feature selection perspective, where we weigh the importance of a token according to how much it can change the loss once masked. If the loss changes greatly after masking a token of interest, it means that such a token has a significant impact on the final decision and is thus relevant. Otherwise, the token is less important for the final decision, so it can be filtered out. After applying the token filtering module generalized from the whole training data, the token number fed to the self-attention module can be obviously reduced in the inference phase, leading to much fewer computations in all the subsequent self-attention layers. The token filter can be realized using a very simple network, where we utilize multi-layer perceptron. Except for the uniqueness of performing token filtering only once from the very beginning prior to self-attention, the other core feature making our method different from the other token filters lies in the predictability of token impact from a feature selection point of view. The experiments show that the proposed method provides an efficient way to approach a light weighted model after optimized with a backbone by means of fine tune, which is easy to be deployed in comparison with the existing methods based on training from scratch.

1. Introduction

Transformer as an emerging model for natural language processing [31] has attracted much attention in computer vision. So far, a couple of vision Transformers have been proposed and made tremendous success in promising superior performance in a variety of applications compared with convolution neural network based deep learning frameworks [7, 29, 3, 1, 18, 32, 13, 23, 34]. At the same time, a major problem arises: The heavy computational load prevents such models from being applied to edge computing-based applications. Therefore, a recent trend has been shifted to develop light weighted models of vision Transformer (ViT). It is known that self-attention is the major bottleneck to incur dense computations in a Transformer as it requires permutation to couple tokens. Accordingly, the recent efforts were devoted to the following trials: (1) Enforce the self-attention to be confined in a neighborhood around each token such that fewer tokens will be involved in updating each token. The methods falling in this category include Swin Transformer [23], Pale Transformer [33], HaloNet [30], and CSWin Transformer [6]. These methods are based on such an assumption that tokens spatially far away are not semantically correlated, but this does not always hold true. Moreover, since the neighborhood to confine self-attention is predefined, not machine learning based, it may sometimes not be coherent to practice. (2) Another solution aims to modify the self-attention operations internally [36, 4, 2, 15]. By changing the computing order in self-attention while incorporating the combination of multiple heads into the self-attention, the complexity of Hydra Attention [2] could be made relatively low provided no non-linear component is contained in the self-attention module, which is a strong constraint to prevent such a solution from being applied broadly. (3) On account of the $O(N^2d)$ complexity of self-attention, where N is the token number and d the feature dimension, a straightforward way is to reduce the number of tokens fed to self-attention instead of the effort to modify self-attention itself. One methodology is to group similar tokens into clusters via unsupervised learning and let each cluster act as a higher-level abstractive representation to take part in the self-attention [39, 20]. Here, the difficulty lies in the quality control of clustering, which may lead to not semantically meaningful representations, and thus affect the final decision negatively. The other kind of solution aims to reduce the token number by applying tokens filter explicitly or based on certain heuristics. In [26], a couple of token filters realized using multi-layer perceptron (MLP) are incorporated into some middle layers of ViT as gating functions, which are trained end-to-end with the

* Corresponding author: suyang@fudan.edu.cn

backbone [29, 14], such that the tokens resulting from one self-attention layer can be selectively forwarded to the subsequent self-attention layers. In [37], an early stop criterion based on the accumulated token value at the first dimension is proposed. In [21], token importance is assumed to be its attentive weight correlated to class token. However, the complex coupling layer by layer brings in uncertainty to the attentive weights in terms of correlating to class token, so gradual token filtering has to be applied while the less attentive tokens are also preserved to aid further testing.

In sum, these token filtering methods miss to address the following issue: They are based on heuristics [37, 21] or enclosed in the end-to-end training with backbone [26], so the rationality of discarding some tokens selectively is not straightforward. In other words, due to the heuristic and less explainable nature of these methods, they are unable to foresee the impact of a token on the final decision explicitly. Therefore it is impossible for them to filter out all irrelevant tokens from the very beginning and token filtering has to be done gradually in a layer-wise manner, which results in unpredictable token filtering on the fly, not favored by parallel computing.

This study aims to solve the aforementioned problem by proposing a ranking method to measure how relevant a token is in regard to the final decision. Based on such a measure, then, we proceed to train a binary classifier as a token filter with learnable parameters generalized from the whole training corpus, such that we can filter out irrelevant tokens from the very beginning prior to self-attention. For this sake, we propose a measure referred to as delta loss (DL) to evaluate how much the loss changes once masking the token of interest, where the naive Transformer can act as the agent to score the difference of loss caused by with or without a token of interest. The mechanism is similar to a wrapper in the sense of classical feature selection [17]. Then, we label the tokens resulting in big DL values as positive instances since masking them will have a significant impact on the final decision. Further, we train a MLP based binary classifier using the labeled tokens based on their DL values. Finally, we apply such a token filter on each token, prior to all the subsequent Transformer blocks, and fine tune the whole pipeline end-to-end. As a result, the irrelevant tokens can be discarded from the very beginning, which is a one-pass process in contrast to reducing token numbers gradually [26, 37, 21].

The contribution of this work is as follows:

(1) In the context of light weighted ViT, it is the first time that token filtering is proposed from a feature selection point of view to rank the relevance of each token in regard to the final decision. Hence, whether a token makes sense for the final decision becomes predictable from the very beginning, which can prevent irrelevant tokens from taking part in self-attention to the best extent. As a one-pass filter deployed at the very beginning prior to self-attention, it can lead to higher efficiency with even fewer token dropout compared with gradual token dropout throughout the pipeline.

(2) We propose a new metric referred to as delta loss to weigh the importance of each token in terms of affecting the final decision and then force the token classifier to optimize its performance on the pseudo labels quantized from the DL values.

(3) The only change compared to the original ViT is applying a MLP as the pre-filter for binary classification, which is fine turned with backbone, so the deployment is quite simple compared with the state-of-the-art (SOTA) methods, which rely on training from scratch.

(4) The experiments show that the proposed method promises SOTA performance in terms of both precision and efficiency in an overall sense.

2. Related works

Vision Transformer. Transformer is initially applied in natural language processing (NLP) [31]. ViT [7] is the first work extending Transformer to computer vision by using no-overlapping image patches for image classification such that no convolution operation is needed. It shows comparable performance to convolution neural networks (CNN) on large-scale datasets. To perform well on various vision tasks, however, ViT and some of its following variants [3, 18, 1] require large-scale data and long training time for pre-training. DeiT [29] improved the training configuration by using a novel distillation method and proposed a Transformer architecture that can be trained only with ImageNet1K [5], whose performance is even better than ViT.

Efficient Transformer. Although Transformer has recently led to great success in computer vision, it suffers from dense computations arising from self-attention, which is also the major mechanism to grant the promising performance in various down-streaming applications. Therefore, recent efforts are focused on proposing various methods to reduce the self-attention caused by dense computations. Provided there are N tokens of d dimension corresponding with the image patches, the self-attention to correlate every couple from the permutation of the N tokens will result in $O(N^2d)$ complexity in a simple updating round. For deploying Transformer on edge devices, a variety of simplified models have been proposed, aiming to reduce parameters and operations, for example, parameter pruning [12, 28], low-rank factorization [38], and knowledge distillation [24, 35]. Yet, these strategies for acceleration are limited in that they still rely on CNN, which deviates from the original design of Transformer, that is, facilitating deep learning with a new working mechanism other than CNN.

One way for rendering light weighted vision Transformer is to simplify the layers of Transformer [40, 25, 8], but its benefit is limited since the major complexity arises

from self-attention, not layer stack. So, some other efforts are focused on altering the internal operations of Transformer to make self-attention more efficient [36, 4, 2]. As for Hydra Attention [2], the computing order insider self-attention is reorganized while the conbination of multiple heads is incorporated into self-attention to reduce the complexity. Nevertheless, it is workable only when no nonlinear component such as SoftMax is applied in self-attention, which limits its applications.

Some other methods try to alleviate the computations of self-attention by reducing the number of tokens. One way is to enforce the computation of self-attention to be conducted in a predefined local region, for instance, Swin Transformer [23], Pale Transformer [33], HaloNet [30], and CSWin Transformer [6]. These methods are based on the assumption that image patches located far from each other are not semantically relevant, but this only partially holds true. Besides, since determining the local context does not rely on machine learning, it cannot be adaptive to various real scenarios end-to-end. Another solution is grouping similar tokens together to obtain more abstractive sparse token representations from clustering. The self-attention confined to such highly abstractive representations can thus be made efficient. TCFormer [39] fuses the tokens in the same cluster into a new one utilizing a weighted average, and the tokens involved in self-attention can then be reduced layer by layer. When tackling high-resolution images, Liang et al. [20] leverage clustering in the first few layers to reduce the number of tokens and reconstruct them in the last few layers. Thus, the dense computations on self-attention can be avoided in the middle layers. The limit for the clustering based methods is: They simply merge similar tokens but ignore the quality control of token clustering in case some clusters might be spanned by less homogeneous tokens.

Since the aforementioned approaches suffer from hard quality control or lack of machine learning, this gives rise to another methodology, which aims to filter out tokens gradually throughout the pipeline of ViT. Dynamic ViT [26] incorporates a couple of learnable neural networks to the middle layers of ViT as the gating structure to make tokens gradually sparser throughout a relatively long course. A-ViT [37] calculates the accumulated halting probability of each token by using the feature values resulting from each Transformer layer, which gradually reduces the number of tokens without adding any additional modules, but could result in suddenly halted computing on a token, in general, not favored when scheduling parallel computing. E-ViT [21] assumes that top-k attentive weights correspond with relevant tokens but it still preserves irrelevant tokens throughout the whole pipeline to undergo a gradual token dropout procedure. The reason is: Token impact cannot be related to final decision in an explicit way due to the complex inter-layer coupling between tokens when back tracing each token's correlation to class token. Besides, every trail of the hyper parameter k in preserving selectively the top-k attentive tokens will lead to a new-round training from scratch. A common limit of the aforementioned approaches is: All such works rely on the running results of the backbone for token filtering, as it is impossible for them to foresee the token-caused effect on the final decision from the very beginning. In view of such a limit, we propose a new method from a feature selection point of view to conduct token filtering from the very beginning prior to self-attention to filter out truly irrelevant tokens.

Feature selection. In the literature on deep learning, Le et al. [19] proposed a feature selector by adding a sparse one-to-one linear layer. It directly uses network weight as the feature weight, so it is sensitive to noise. Roy et al. [27] used the activation potential as a measure for feature selection at each single input dimension but is limited to specific DNNs. Since then, the interest has been turned to the data with a specific structure, which relies more on the progress of traditional data feature selection methods [10, 22]. AFS [11] proposes to transform feature weight generation into a mode that can be solved by using an attention mechanism. Takumi et al. [16] proposed a method that harnesses feature partition in SoftMax loss function for effectively learning the discriminative features. However, these methods are focused on reducing feature map or selecting channels of CNN rather than Transformer. We are the first to use the delta loss value as an indicator for identifying relevant Transformer tokens from a wrapper-based feature selection point of view [17] by testing their impact on the final decision once masked.

3. DL-ViT

We propose a metric referred to as delta loss to weigh how vital a token is. In detail, we mask a token at first and then compute its impact on the loss, say, the change of cross entropy with/without such token for the final decision. If masking a token leads to big DL, it means that such a token does affect the final decision much, which should be preserved to take part in the subsequent self-attentions. Vice versa, if the loss does not change much with/without a token, such a token should be discarded due to its less importance to the decision. Correspondingly, a plausible trick arising from the aforementioned scheme is: The Transformer itself can act as the agent to score the importance of each token via DL without any further machine learning required in this phase. By using the DL scores to label the tokens in the training corpus as positive or negative, we can then train a binary classifier to check whether the tokens of an input image should be preserved to take part in the subsequent self-attentions, where the classifier is implemented using MLP. Finally, we preset the simple MLP module prior to the backbone Transformer, and fine tune

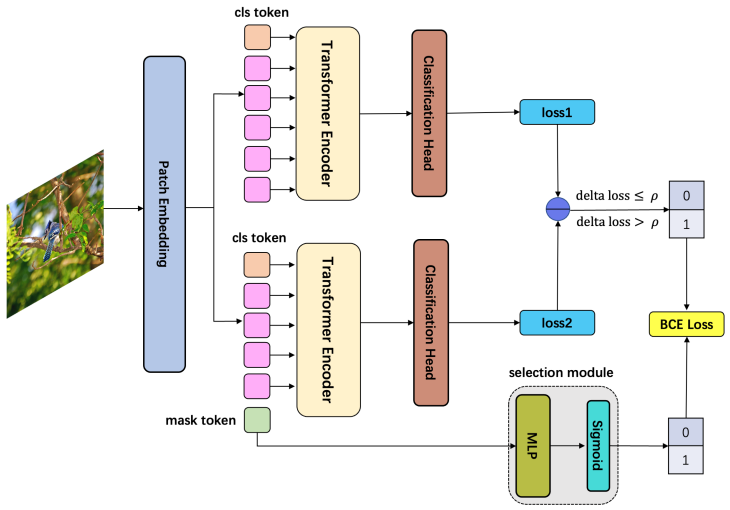

Figure 1. The overall training process: The two branches of the vision Transformer are in fact the same one, whose parameters are fixed during training. The delta loss and ρ refer to Eq. (7) and Eq. (8).

the whole pipeline, where preset a token filter as such is the only change in the architecture.

In the following, we describe the two phases of the delta loss based efficient vision Transformer (DL-ViT): Evaluating token importance with delta loss to train the token filter and then fine tuning the entire network after incorporating the token filter. After an image passes through the embedding layer of the vision Transformer, the non-overlapping image patches are encoded into tokens denoted as:

$$X = \left\{ x_i \in \mathcal{R}^d | i = 1, 2 \ldots, N \right\}, \tag{1}$$

where N is the total number of tokens and d the embedding dimension. After masking the i-th token $x_i \in \mathcal{R}^d$, we get the tokens in the following form:

$$X_i = \left\{ x_1, \ldots x_{i-1}, \varnothing, x_{i+1}, \ldots, x_N \right\}, \tag{2}$$

where $\varnothing$ means replacing the i-th token with zeros (masking). Then, we feed X and X_i to the Transformer, respectively, to obtain the corresponding prediction results:

$$\hat{y} = Transformer(X), \tag{3}$$

$$\hat{y}_i = Transformer(X_i). \tag{4}$$

Based on the previous prediction results, we calculate the cross-entropy loss of either case in reference to the ground truth y as follows:

$$\mathcal{L} = CrossEntropy(\hat{y}, y), \tag{5}$$

$$\mathcal{L}_i = CrossEntropy(\hat{y}_i, y). \tag{6}$$

It is known that the value of loss measures how close the prediction result approaches the ground truth, where a lower value corresponds with closer to the ground truth. Let:

$$\Delta\mathcal{L}_i = \mathcal{L} - \mathcal{L}_i. \tag{7}$$

Obviously, if the delta loss defined in Eq. (7) is positive, it means that masking the i-th token makes the decision closer to the ground truth since masking as such causes a lower cross-entropy value in contrast to the original case. In such a case, discarding the token should not affect but benefit the decision of ViT, and a bigger delta loss corresponds with a better change on the decision. So, we quantize the delta loss measure to mark whether the current token should be discarded or not, formulated as:

$$label(x_i) = \begin{cases} 0, & \mathcal{L} - \mathcal{L}_i \leq \rho \\ 1, & \mathcal{L} - \mathcal{L}_i > \rho \end{cases} \tag{8}$$

where 0 means leaving the token out, 1 preserving the token, and ρ the only hyperparameter to control the significance of the pseudo labeling.

After labeling all the tokens in the training corpus, we can then proceed to learn the generalizable law to distinguish positive token examples from negative ones in a population sense, which leads to a binary classifier realized using MLP for token filtering, acting to determine whether each token should be preserved to the next phase of the pipeline or not. So far, there is still a critical problem to be tackled, that is, some similar tokens may lead to contradicting results in terms of delta loss. This is quite common when two images share some similar patches locally but are quite different in an overall sense. Such semantically ambiguous local patches impose difficulty on token filter training, so we attach the profile featuring the whole image to each token as context, namely, global feature, to solve this problem. That is, we not only use the tokens with original embedding but also apply adaptive average pooling (AAP) over all tokens of an image to obtain the global feature of the image, acting as the context to make each token distinguishable from the

others. Thus, the overall descriptor for each token becomes:

$$x_i' = [x_i, x_{global}]. \tag{9}$$

$$x_{global} = AAP(X) = \frac{1}{N}\Sigma_{k=1}^{N} x_k. \tag{10}$$

Consequently, x_i' instead of x is fed to the token selection module for training and inference:

$$p_i = Sigmoid(MLP(x_i')). \tag{11}$$

During training, we first fix all parameters of the pre-trained backbone Transformer for token labeling, and then, train the MLP only. Here, we use binary-cross-entropy loss to train the network:

$$\mathcal{L}_{MLP} = BinaryCrossEntropy(p_i, label(x_i)), \tag{12}$$

where p_i is the prediction from MLP, and $label(x_i)$ the pseudo label calculated from Eq. (8). Fig. 1 depicts how to train the token selection module with delta loss. Algorithm 1 and Algorithm 2 describe respectively how to label tokens with naive DeiT [29] and how to train the selection module.

Algorithm 1 Token labeling with naive DeiT [29]

Input: $\mathbf{X} = \{x_i \in \mathcal{R}^d | i = 1, 2..., N\}$, and the corresponding ground truth y.
Output: **Label** $= \{label(x_i) | i = 1, 2..., N\}$.

Label $= \varnothing$
Set ρ to control the significance of pseudo labeling.
$\hat{y} = Transformer(X)$
$\mathcal{L} = CrossEntropy(\hat{y}, y)$
for $i = 1, 2, ..., N$ **do**
  $X_i = \{x_1, ...x_{i-1}, \varnothing, x_{i+1}, ..., x_N\}$
  $\hat{y}_i = Transformer(X_i)$
  $\mathcal{L}_i = CrossEntropy(\hat{y}_i, y)$
  if $\mathcal{L} - \mathcal{L}_i \leq \rho$ **then**
    $label(x_i) = 0$
  else
    $label(x_i) = 1$
  end if
  Label $=$ **Label** $\cup\, label(x_i)$
end for
return Label

As shown in Fig. 2, before entering the Transformer, all tokens must go through the token selection module that will output the decision of keeping or discarding the token. During fine tuning, we train both the token selection module and the DeiT end-to-end, based on the cross-entropy loss:

$$\mathcal{L}_{finetune} = CrossEntropy(\hat{y}, y), \tag{13}$$

Algorithm 2 Token filter training

Input: Batch of images with tokens and the corresponding pseudo labels in the form of $\mathbf{X} = \{x_i \in \mathcal{R}^d | i = 1, 2..., N\}$ and **Label** $= \{label(x_i) | i = 1, 2..., N\}$.
Output: Parameters **W** of the MLP token filter.

Random initialization of **W**
repeat
  Load **(X,Label)** of one image in the batch in turn
  $x_{global} = AAP(X)$
  for i = 1, ..., N **do**
    $x_i' = [x_i, x_{global}]$
    $p_i = Sigmoid(MLP(x_i'))$
    $\mathcal{L}_{MLP} = BinaryCrossEntropy(p_i, label(x_i))$
    Back-propagation updating **W**
  end for
until no more descent on L_{MLP}
return W

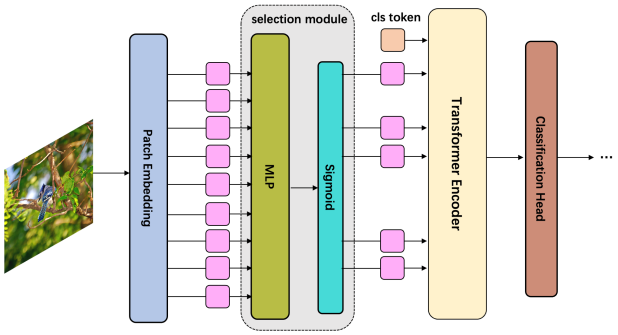

Figure 2. The overall fine tuning and inference process of the proposed approach. All the tokens enter the selection module in turn to decide whether they should be passed to the subsequent pipeline of Transformer according to the predicted probability, after which the number of the preserve tokens will remain unchanged in the rest pipeline.

where y is the ground truth and $\hat{y}$ the output of the whole network.

During fine tuning, in order to make it easy to parallelize the computation, we do not delete tokens directly but replace them with zeros to prevent them from affecting subsequent operations. Such a token masking strategy makes the computational cost of the training iterations similar to those of the original vision Transformer. During inferring, we throw the masked tokens out of the subsequent calculations in order to examine the actual acceleration resulting from the token selection mechanism.

4. Experiments

Data: We evaluate our method for image classification on the 1000-class ImageNet1K ILSVRC 2012 dataset [5],

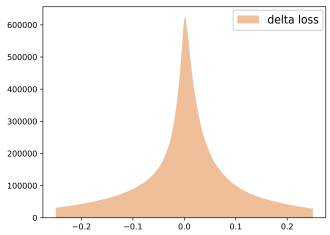

Figure 3. Distribution of all the DL values on ImageNet1K training set.

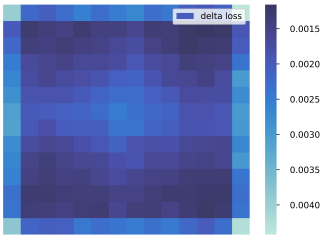

Figure 4. Average DL of every image patch obtained from DeiT-T [29]. Darker color corresponds with lower DL value.

and all images have a resolution of 224×224.

Experimental setting: Following the baselines [37, 26, 21], we use the data-efficient vision Transformer (DeiT) [29] as the backbone, and following its training principles, we only use the ImageNet1K dataset for training. We use 16×16 patch resolution and SGD optimization. The MLP is composed of 3 layers with ReLU for the first two layers and Sigmoid for the last layer as the activation and the number of neurons are set to 384, 100, and 1 for each layer, respectively. When training MLP, we use the pre-trained model of DeiT to compute the loss value, and the learning rate is fixed to 1×10^{-2}. When fine tuning the whole network, the learning rate is 1×10^{-3} and reduced by 10 times for every 40 epochs. For regularization, we set the weight decay of the optimizer to 1×10^{-4} in both MLP training and fine tuning. Starting from publicly available pre-trained checkpoints and the pre-trained token selection module, we fine tune the DL-ViT-T/S variant models evolved from DeiT-T/S [29] for 100 epochs, where T/S refers to 3-head/6-head with 192-dimension/384-dimension implementation on 12 layers, respectively. We use 2 NVIDIA 3090 GPUs for training.

4.1. Intuitive insight from statistics

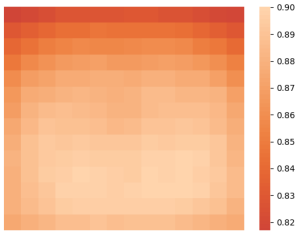

Figure 5. The average of masks predicted by our token selection module on ImageNet1K validation set.

Distribution of DL values: In order to examine the rationality of our method intuitively, we visualize the distribution of all DL values in Fig. 3. We find that small DL values around 0 dominate the majority of the distribution, which reveals the fact that only a part of the image patches are significantly relevant to classification. So considerable tokens with smaller DL values can be discarded.

Fig. 4 depicts the average DL value of the tokens at each patch resulting from DeiT-T [29] on the training set of ImageNet1K. We find that most semantically important patches are on the center of an image, the rest of which should fall into the outliers to be eliminated more frequently by our method.

Qualitative analysis. Fig. 5 visualizes the masks on the ImageNet1K validation set resulting from DL-ViT, where the dark portion appearing mostly along the edge of an image are the less contributive patches for classification, namely, the outliers favored by the token filter to activate elimination. Sometimes, the token selection module eliminates not only the background of the image, but also the confusing portion that may cause classification errors. For example, the third image in the last row of Fig. 6 prefers eliminating the patches unrelated to the dog.

4.2. Comparison to baselines

We compare our method with the baselines in Table 1 in terms of efficiency and precision, where we set ρ to 0.002 and 0.001 for DL-ViT-T and DL-ViT-S, respectively. At the cost of sacrificing only 0.3% and 0.2% accuracy compared with that of the backbone, we cut down 46% and 15% FLOPs of DeiT-T and DeiT-S, respectively. Moreover our method performs best to make DeiT-T more efficient and more precise compared with the baselines, where the Floating-point Operations (FLOPs) metric is measured by FlopCountAnalysis[9].

As E-ViT is an exception that only reports comparison on ViT-S, we follow A-ViT [37] and Dynamic-ViT [26] to report the performance on both ViT-S and ViT-T. Regarding ViT-S, no method performs best on all metrics, where E-ViT runs faster but is inferior to DL-ViT on top-1 precision. Except for the highest top-1 precision on both benchmarks, DL-ViT promises the state-of-the-art (SOTA) performance in an overall sense if taking into account both benchmarks. Since E-ViT misses to compare with all the baselines on ViT-T, except for the performance, we compare it with DL-ViT in a methodological sense to allow a more comprehensive insight: (1) We evaluate token importance via delta loss while E-ViT leverages top-k attention

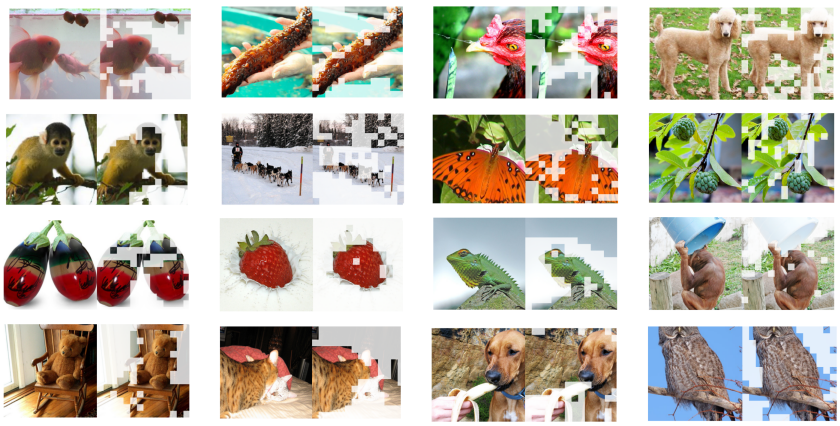

Figure 6. Original image (left) and the masked image (right) resulting from DL-ViT on the ImageNet1K set. The left two columns are the validation set, and the right two columns the training set.

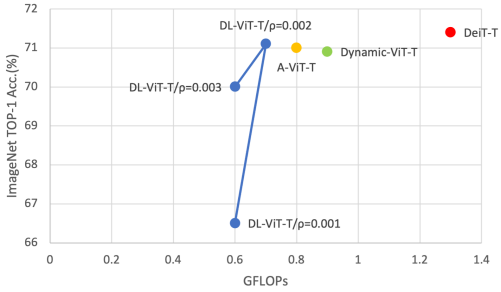

Figure 7. The tiny model complexity (FLOPs) and top-1 accuracy trade-offs on ImageNet.

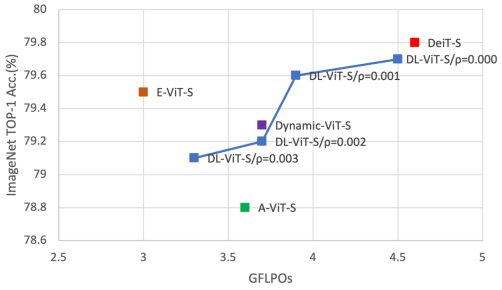

Figure 8. The small model complexity (FLOPs) and top-1 accuracy trade-offs on ImageNet.

weights as token importance; (2) We filter out irrelevant tokens from the very beginning but E-ViT does this gradually and preserve both important and less important tokens in the whole pipeline. That is, E-ViT cannot foresee the impact of each token on the final decision at the beginning but DL-ViT can. (3) E-ViT modifies the self-attention, and the whole pipeline has to be changed wherever attention is applied, so it has to train from scratch, which is too expensive compared with the fine tune as adopted in our framework. As we change nothing in ViT, the pseudo labeling is performed by using naive ViT without any training. Besides, MLP is a two-class classifier, whose training is not tough. In this sense, the change on the architecture is minor. (4) In DL-ViT, ρ controls the significance of pseudo labeling, where the heuristics to choose its value lies in the statistics of DL values as shown in Fig. 3. For E-ViT, determining k is not easy in that there is no explicit heuristic to foresee its impact on the overall performance, and every trail will lead to a new-round computation-intensive training from scratch. Besides, the layer-varying token importance accounts for why layer-wise token dropout has to be done gradually. In Fig. 7 and Fig. 8, we compare DL-ViT with the baselines under different settings. It is obvious that our model can achieve a good trade-off between efficiency and precision.

In addition to FLOPs, we also evaluate the image throughput of our model on a single NVIDIA RTX 3090 GPU with batch size fixed to 64, and for GPU warming up, 512 forward passes are conducted. The experiment demonstrates that our DL-ViT can accelerate the inference by $15\% \sim 41\%$.

4.3. Ablation study

In our method, the full configuration of a solution is subject to the following factors: The backbone for token importance evaluation, the threshold ρ to control the annotation on DL values, MLP, and the local/global feature applied to it. As shown in Table 2, a high threshold value of ρ can filter out more tokens, resulting in higher efficiency, but a too high one will cause degradation in precision. So, there is a compromise to determine the value of ρ, where we let $\rho = 0.002$ for DL-ViT-T. Note that when $\rho = 0.002$, the accuracy of using only local features is even higher than that of DeiT-T [29], at the cost of sacrificing FLOPs. Yet, we incorporate global feature as our primary solution due to its promising overall performance. Note that both cases lead to varying accuracy when ρ changes from 0.001 to 0.003, but

Model	Efficiency			Top1 Acc.(%) ↑	Resolution
	#Params.↓	FLOPs ↓	Throughput ↑		
ViT-B [7]	86.0M	17.6G	563 imgs/s	77.9	224
DeiT-S [29]	22.0M	4.6G	1500 imgs/s	79.8	224
Dynamic-ViT-S [26]	22.7M	3.7G	1654 imgs/s	79.3	384
A-ViT-S [37]	**22.0M**	3.6G	1849 imgs/s	78.8	224
E-ViT-S [21]	22.1M	**3.0G**	**1923 imgs/s**	79.5	224
DL-ViT-S(ours)	22.1M	3.9G	1602 imgs/s	**79.6**	224
DeiT-T [29]	5.7M	1.3G	3231 imgs/s	71.4	224
Dynamic-ViT-T [26]	5.9M	0.9G	4361 imgs/s	70.9	224
A-ViT [37]	5.7M	0.8G	4523 imgs/s	71.0	224
DL-ViT-T(ours)	**5.7M**	**0.7G**	**4565 imgs/s**	**71.1**	224

Table 1. Comparison with baselines. Except for E-ViT, which undergoes training of 300 epochs, the other models are trained with 100 epochs. Note that Dynamic-ViT-S turns out from the resolution of 384 × 384.

Threshold	Top-1 Acc. (%)↑	Top-5 Acc. (%)↑	FLOPs ↓	Throughput (images/s)↑	Top-1 Acc. (%)↑	Top-5 Acc. (%)↑	FLOPs ↓	Throughput (images/s)↑
DeiT-T [29]	71.4	90.8	1.3G	3231	71.4	90.8	1.3G	3231
	DL-ViT-T with local feature				DL-ViT-T with local and global feature			
0.001	72.0	90.8	0.8G	4771	66.5	86.1	0.6G	4690
0.002	**73.1**	**91.5**	1.0G	3937	**71.1**	**89.9**	0.7G	4565
0.003	62.4	83.5	**0.3G**	**5996**	70.0	89.1	**0.6G**	**5527**

Table 2. Performance of DL-ViT-T subject to local/global feature and threshold ρ.

Strategy	Metric		
	#Params.↓	FLOPs ↓	Top1 Acc.(%) ↑
DL-ViT-T	**5.7M**	0.7G	**71.1**
DeiT-T_0 [29]	5.7M	0.7G	68.2
DL-ViT-T_0	5.7M	**0.4G**	56.6

Table 3. Comparison with DeiT using random token discard (DeiT-T_0 in the second row) and DL-ViT without pre-training but randomly initializing the MLP (DL-ViT-T_0 in the third row).

the FLOPs with global feature remain stably low.

Table 3 shows that the proposed model degrades in terms of precision if replacing the pre-training of MLP with the random initialization, and the backbone based on random token filtering also leads to inferior performance. This indicates that our token filtering scheme does contribute to making the DeiT-T efficient while preserving its precision to the best extent.

5. Conclusions

We develop an efficient vision transformer with token impact prediction such that token filtering can be deployed at the very beginning prior to self-attention, where the backbone Transformer is used as an agent/wrapper to rank the impact in terms of the difference of loss caused by masking a token of interest. It is the first time to develop a light-weighted model from a feature selection point of view with explicit insight into token's relevance to the decision. A MLP for token filtering is the only added module, which acts as a two-class classifier with minor change on the overall architecture, and its training is not tough. The present solution is a one-pass filter. In the future, we will investigate into the relevance of tokens at middle layers to the final decision to further improve the efficiency.

References

[1] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 1, 2

[2] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. Hydra attention: Efficient attention with many heads. *arXiv preprint arXiv:2209.07484*, 2022. 1, 3

[3] 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. 1, 2

[4] 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. 1, 3

[5] 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. 2, 5

[6] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12124–12134, 2022. 1, 3

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 8

[8] Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. *arXiv preprint arXiv:1910.10073*, 2019. 2

[9] facebook research. https://github.com/facebookresearch/fvcore, 2021. 6

[10] Nina Grgić-Hlača, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. Beyond distributive fairness in algorithmic decision making: Feature selection for procedurally fair learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 3

[11] Ning Gui, Danni Ge, and Ziyin Hu. Afs: An attention-based mechanism for supervised feature selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3705–3713, 2019. 3

[12] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1389–1397, 2017. 2

[13] Drew A Hudson and Larry Zitnick. Generative adversarial transformers. In *International conference on machine learning*, pages 4487–4499. PMLR, 2021. 1

[14] Zihang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Xiaojie Jin, Anran Wang, and Jiashi Feng. Token labeling: Training a 85.4% top-1 accuracy vision transformer with 56m parameters on imagenet. 2021. 2

[15] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *Learning*, 2020. 1

[16] Takumi Kobayashi. Group softmax loss with discriminative feature grouping. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2615–2624, 2021. 3

[17] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997. 2, 3

[18] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509, 2022. 1, 2

[19] Yifeng Li, Chih-Yu Chen, and Wyeth W Wasserman. Deep feature selection: theory and application to identify enhancers and promoters. *Journal of Computational Biology*, 23(5):322–336, 2016. 3

[20] Weicong Liang, Yuhui Yuan, Henghui Ding, Xiao Luo, Weihong Lin, Ding Jia, Zheng Zhang, Chao Zhang, and Han Hu. Expediting large-scale vision transformer for dense prediction without fine-tuning. *arXiv preprint arXiv:2210.01035*, 2022. 1, 3

[21] 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. *arXiv preprint arXiv:2202.07800*, 2022. 2, 3, 6, 8

[22] Hongfu Liu, Haiyi Mao, and Yun Fu. Robust multi-view feature selection. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 281–290. IEEE, 2016. 3

[23] 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, 3

[24] 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. 2

[25] Zizheng Pan, Bohan Zhuang, Haoyu He, Jing Liu, and Jianfei Cai. Less is more: Pay less attention in vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2035–2043, 2022. 2

[26] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *NeurIPS*, 34:13937–13949, 2021. 1, 2, 3, 6, 8

[27] Debaditya Roy, K Sri Rama Murty, and C Krishna Mohan. Feature selection using deep neural networks. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2015. 3

[28] Maying Shen, Pavlo Molchanov, Hongxu Yin, and Jose M Alvarez. When to prune? a policy towards early structural pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12247–12256, 2022. 2

[29] 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, 6, 7, 8

[30] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021. 1, 3

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia

Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 2

[32] 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. 1

[33] Sitong Wu, Tianyi Wu, Haoru Tan, and Guodong Guo. Pale transformer: A general vision transformer backbone with pale-shaped attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2731–2739, 2022. 1, 3

[34] Enze Xie, Wenhai Wang, Zhiding Yu, Animashree Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *neural information processing systems*, 2021. 1

[35] Pengtao Xie and Xuefeng Du. Performance-aware mutual knowledge distillation for improving neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11922–11932, 2022. 2

[36] Chenglin Yang, Yilin Wang, Jianming Zhang, He Zhang, Zijun Wei, Zhe Lin, and Alan Yuille. Lite vision transformer with enhanced self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11998–12008, 2022. 1, 3

[37] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *CVPR*, pages 10809–10818, 2022. 2, 3, 6, 8

[38] Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. On compressing deep models by low rank and sparse decomposition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7370–7379, 2017. 2

[39] Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11101–11111, 2022. 1, 3

[40] Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. Bert loses patience: Fast and robust inference with early exit. *Advances in Neural Information Processing Systems*, 33:18330–18341, 2020. 2