Adaptive Sparse ViT: Towards Learnable Adaptive Token Pruning by Fully Exploiting Self-Attention

Xiangcheng Liu[1] , **Tianyi Wu**[2*] and **Guodong Guo**[3†]
[1]Peking University
[2]Baidu Autonomous Driving Technology Department (ADT)
[3]Institute of Deep Learning, Baidu Research
liuxiangcheng@stu.pku.edu.cn, wutianyi01@baidu.com, Guodong.Guo@mail.wvu.edu

Abstract

Vision transformer has emerged as a new paradigm in computer vision, showing excellent performance while accompanied by expensive computational cost. Image token pruning is one of the main approaches for ViT compression, due to the facts that the complexity is quadratic with respect to the token number, and many tokens containing only background regions do not truly contribute to the final prediction. Existing works either rely on additional modules to score the importance of individual tokens, or implement a fixed ratio pruning strategy for different input instances. In this work, we propose an adaptive sparse token pruning framework with a minimal cost. Specifically, we firstly propose an inexpensive attention head importance weighted class attention scoring mechanism. Then, learnable parameters are inserted as thresholds to distinguish informative tokens from unimportant ones. By comparing token attention scores and thresholds, we can discard useless tokens hierarchically and thus accelerate inference. The learnable thresholds are optimized in budget-aware training to balance accuracy and complexity, performing the corresponding pruning configurations for different input instances. Extensive experiments demonstrate the effectiveness of our approach. Our method improves the throughput of DeiT-S by 50% and brings only 0.2% drop in top-1 accuracy, which achieves a better trade-off between accuracy and latency than the previous methods.

1 Introduction

Recently, Vision Transformer (ViT) has made remarkable progress on image classification [Dosovitskiy *et al.*, 2020; Touvron *et al.*, 2021a; Liu *et al.*, 2021], object detection [Carion *et al.*, 2020; Zhu *et al.*, 2020], semantic segmentation [Zheng *et al.*, 2021; Xie *et al.*, 2021], and other vision tasks. However, as the model complexity is quadratic to the number of tokens, ViT suffers from expensive computational costs, which limits its application and deployment.

Not all image patches are helpful for the final prediction. For instance, the large number of image tokens in the background region do not contribute to the recognition and can be pruned during the inference process, greatly accelerating the model runtime without significant impact on performance. Token pruning has attracted the interests of many researchers. We classify the existing methods according to whether additional calculations are introduced to evaluate the token score. Evo-ViT [Xu *et al.*, 2022] utilizes class attention [Xu *et al.*, 2022] to estimate token score and develop a novel slow-fast token evolution approach to improve the throughput of ViT. EViT [Liang *et al.*, 2022] employ a similar method to measure token importance and fuse discarded tokens. Both of these approaches require manually specifying the pruning ratio for each stage, and perform the same pruning policy for different input instances, which may result in simple samples being under-pruned or complex samples being over-pruned in the beginning stages, as illustrated in Figure 2 top. Another type of work identifies token importance via extra measures. DynamicViT [Rao *et al.*, 2021] prune tokens in a fixed ratio by inserting lightweight predictors to predict token scores. IA-RED^2 [Pan *et al.*, 2021] introduce a multi-head interpreter and employed reinforcement learning to generate pruning scheme for each token. A-ViT [Yin *et al.*, 2021] compute halting scores for all tokens to adaptively discard unimportant tokens. The latter two achieve sample-adaptive pruning at the cost of additional computation of significance scores for all tokens.

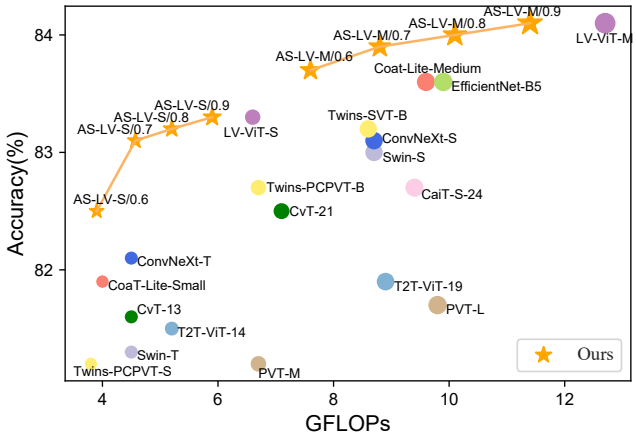

Figure 1: Trade-offs between complexity and top-1 accuracy for different models on ImageNet.

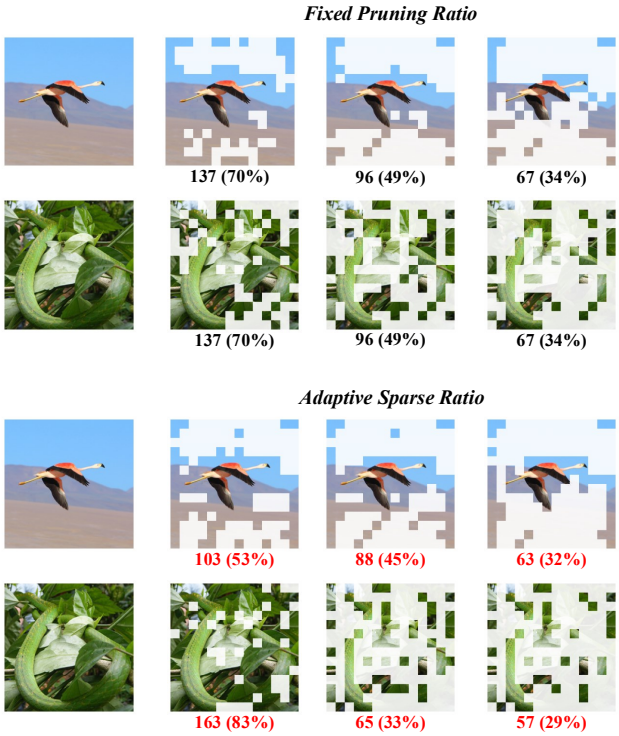

Figure 2: Comparison of fixed pruning rate (up) and adaptive sparse rate (down). The number denotes the amount and percentage of tokens kept in the current stage.

In this work, we propose an adaptive token sparse framework for ViT acceleration, named AS-ViT, which fully exploits Multi-Head Self-Attention (MHSA) to estimate token importance scores, and uses a minimal cost, only three learnable thresholds, to perform the corresponding token pruning policy for a specific input instance as shown in Figure 2 bottom. Specifically, we first propose the attention head importance weighted class attention score. It uses the intermediate results of MHSA to calculate token-level head importance, which is then multiplied as a weighting factor on the class attention [Xu *et al.*, 2022] score to better identify informative tokens. Then, we insert three learnable thresholds in ViT hierarchically. Only tokens with a score greater than the current threshold will be kept, and the pruned token will not participate in the later computations. Lastly, we propose a budget-aware loss to optimize the thresholds to achieve a trade-off between accuracy and computational effort across the dataset. During testing, the learnable threshold is fixed and we just compare the threshold and individual token scores to actually discard uninformative tokens for hardware acceleration. We investigate the intrinsic features of MHSA and reuse its computational results to evaluate the informativeness of the token. Threshold-based comparison avoids sorting all tokens. We use the above method to minimize the cost of token pruning.

We conduct extensive token pruning experiments for the widely used vision transformer backbones, DeiT [Touvron *et al.*, 2021a] and LV-ViT [Jiang *et al.*, 2021] on ImageNet. For instance, our method improves 1.5x throughput with only 0.2% decrease in accuracy while reducing the 35% GFLOPs of the DeiT-small model. For other model and pruning rates, our method also achieves a better accuracy and complexity balance compared with previous approaches.

2 Related Work

Vision Transformer. Transformer [Vaswani *et al.*, 2017], developed in NLP, has been successfully applied to various vision tasks and tends to replace CNN [He *et al.*, 2016; Tan and Le, 2019] gradually. ViT [Dosovitskiy *et al.*, 2020] illustrates that transformer lacks inductive bias and requires large-scale dataset pre-training to achieve an approximate performance with the state-of-the-art CNN. DeiT [Touvron *et al.*, 2021a] eliminates the above problem with well-tuned training parameters and the introduction of distillation token. LV-ViT [Jiang *et al.*, 2021] explores a variety of techniques for training vision transformer, significantly improving the performance of ViT. PVT [Wang *et al.*, 2021] constructs a hierarchical ViT similar to CNN and proposes spatial-reduction attention to reduce the computation complexity. Swin Transformer [Liu *et al.*, 2021] proposes a window based self-attention that makes the complexity linear with respect to token number and becomes a generic visual backbone. DGT [Liu *et al.*, 2022a] propose the dynamic group attention to accelerate inference.

Static ViT Pruning. Analogous to weights pruning in CNN [Han *et al.*, 2015; Li *et al.*, 2016; Molchanov *et al.*, 2016; Liu *et al.*, 2017; Luo *et al.*, 2017; Frankle and Carbin, 2018], there are many works [Zhu *et al.*, 2021; Yang *et al.*, 2021; Chen *et al.*, 2021; Yu *et al.*, 2022] for ViT parameters compression. VTP [Zhu *et al.*, 2021] prunes parameters in MHSA and FFN indiscriminately through L1 regularized sparse training. NViT [Yang *et al.*, 2021] establishes the global weights importance via performing Taylor expansion to the loss, then conducts structured pruning and parameter reassignment based on dimensional trends. SViTE [Chen *et al.*, 2021] fully explores the sparsity of ViT, compressing transformer with structured pruning, unstructured sparsity and token pruning. In this paper, we focus on token compression, which is orthogonal to static ViT pruning, and we can further improve the compression rate, benefit from weights pruning.

Dynamic ViT Pruning. Thanks to the transformer's parallel computing mechanism, pruning image tokens can bring real acceleration without the need of additional support. Both DynamicViT [Rao *et al.*, 2021] and IA-RED^2 [Pan *et al.*, 2021] dynamically keep informative tokens by inserting prediction modules. PS-ViT [Tang *et al.*, 2021] belongs to static token pruning, which statistically obtains the importance distribution of tokens across the dataset and prunes them top-down. EViT [Liang *et al.*, 2022] and Evo-ViT [Xu *et al.*, 2022] use the class attention score to distinguish how informative a token is. A-ViT [Yin *et al.*, 2021] adaptively calculates the halting score for each token. Our method strives to implement instance-wise token pruning without additional computational costs.

3 Method

3.1 Preliminary

Vision Transformer [Dosovitskiy *et al.*, 2020] provides a new paradigm for image recognition. ViT first splits the image into $N \times N$ non-overlapping patches and embeds them into a D dimensional feature space, and then adds a class token before the image tokens for final classification. Considering the position relationship, all tokens are added with a learnable position encoding and then fed into a stacked transformer block. We summarize the operations inside the block by using the following two equations:

$$x_{\text{MHSA}} = x + \text{MHSA}(\text{LN}(x)), \tag{1}$$

$$x_{\text{FFN}} = x_{\text{MHSA}} + \text{FFN}(\text{LN}(x_{\text{MHSA}})), \tag{2}$$

where MHSA stands for Multi-Head Self-Attention [Vaswani *et al.*, 2017], FFN is feed-forward neural network, and LN stands for layer normalization.

MHSA represents features in multiple subspaces using the same parameters. The input is first projected into three matrices Query $\mathbf{Q} \in \mathbb{R}^{(N+1)\times D}$, Key $\mathbf{K} \in \mathbb{R}^{(N+1)\times D}$, and Value $\mathbf{V} \in \mathbb{R}^{(N+1)\times D}$, respectively, and then sliced into H attention heads to perform the parallel operations:

$$\text{Context}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_h}}\right)\mathbf{V}, \tag{3}$$

where $D_h = \frac{D}{H}$ is the feature dimension of the single head output. In particular, the attention scores of class token x_{cls} and other tokens can be written as:

$$\text{A}(x_{\text{cls},:}) = \text{Softmax}\left(\frac{\mathbf{Q}_{\text{cls}}\mathbf{K}^T}{\sqrt{D_h}}\right) \in \mathbb{R}^{H\times N}, \tag{4}$$

which is used to reflect which tokens are contributing to the classification. Next, MHSA concatenates the Context of all attention heads together and project them through a matrix.

3.2 Class Attention Score Weighted by Attention Head Importance

Popular metrics for evaluating the importance of a token include its similarity score to the class token [Liang *et al.*, 2022; Xu *et al.*, 2022] and the attention it receives from other tokens [Goyal *et al.*, 2020; Kim *et al.*, 2021]. However, we observed that both approaches ignore the diversity of attention heads, and the scores of different heads are treated equally, in other words, the final score is their average value.

Considering that different tokens may receive diverse attention in multiple attention heads, we propose a metric to estimate the token-level head importance, and it simply relies on the intermediate results of MHSA. Taking the i-th input token x_i as an example, firstly, we define the Context of the h-th head of the l-th layer as $\text{Context}^{(h,l)}(x_i) \in \mathbb{R}^{D_h}$, then we calculate the l_2-norm along the last dimension as the importance $\mathcal{H}^{(h,l)}(x_i)$ of the h-th head like Equation (5):

$$\mathcal{H}^{(h,l)}(x_i) = \sqrt{\sum_{j=1}^{D_h} \text{Context}_j^{(h,l)}(x_i)^2}. \tag{5}$$

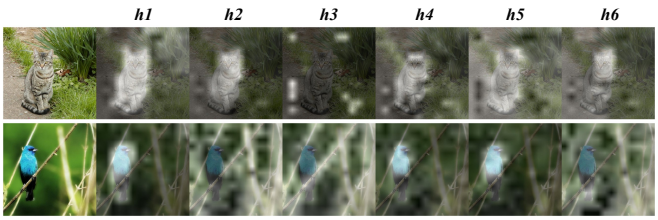

Figure 3: Visualization of tokens' attention paid to each head. The figure on the top is the token-level attention head importance visualization from AS-DeiT-S layer 7, and the figure below is from AS-DeiT-S layer 9.

Our motivation for estimating head importance derives from CNN pruning, of which l_2-norm [He *et al.*, 2018] is a commonly used metric for filter significance rating. In addition, our metrics are token-level, different from the previous head pruning work [Michel *et al.*, 2019]. Next, we take the proportion of the importance of the h-th attention head to the sum of all head as its weighting factor like Equation (6):

$$\mathcal{R}^{(h,l)}(x_i) = \frac{\mathcal{H}^{(h,l)}(x_i)}{\sum_{h=1}^{H}\mathcal{H}^{(h,l)}(x_i)}, \tag{6}$$

$$\mathcal{S}^l(x_i) = \sum_{h=1}^{H}\mathcal{R}^{(h,l)}(x_i)\cdot \text{A}^{(h,l)}(x_{\text{cls,i}}), \quad i = 1, 2, ..., N. \tag{7}$$

We visualize the token-level head importance in the Figure 3 and brighter areas indicate that the current attention head is more important for this token. Taking the second image as an example, the foreground tokens focus more on head 1, 4 and 5, while the background tokens clearly favor head 2 and 3. It is clear that the diversity of individual heads should be taken into account when scoring tokens.

The vanilla class attention score $\text{A}^{(h,l)}(x_{\text{cls,i}})$ of the i-th token is chosen as the basic evaluation metric, and we can estimate token score $\mathcal{S}^l(x_i)$ more accurately through head importance weighting as Equation (7). The flow of head importance weighted class attention score is shown in Figure 4. Compared to existing adaptive token pruning works, our approach takes into account the attention head diversity and has no reliance on any extra scoring mechanisms.

3.3 Adaptive Token Pruning Based on Learnable Thresholds

In order to achieve sample-adaptive token pruning while minimizing the operation cost. We introduce three stage-wise learnable thresholds to control token sparse behavior, following LTP [Kim *et al.*, 2021]. Figure 4 illustrates the overall framework of our proposed method. We usually divide the successive transformer blocks into four stages and insert a learnable threshold θ before the second, third and fourth stages, respectively. Adaptive Sparse Module is a comparator based on thresholds and token scores. We keep tokens with scores greater than the threshold, as in Equation (8):

$$M^n(x_i) = \begin{cases} 1, & \text{if } \mathcal{S}^l(x_i) > \theta^n \\ 0, & \text{otherwise} \end{cases}, \quad n = 1, 2, 3, \tag{8}$$

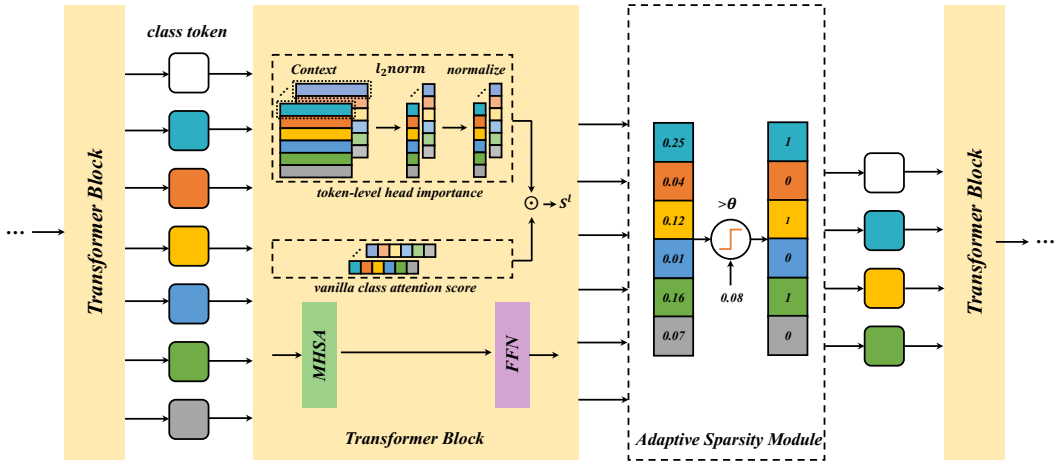

Figure 4: The framework of our Adaptive Sparse ViT. θ denotes the learnable threshold and $\odot$ indicates Hadamard product. The token evaluation metric accounts for the attention of class token as well as the importance of different attention heads. Adaptive Sparse Module fulfills pruning by comparing token scores and thresholds.

where $M^n(x_i)$ is a binary mask to indicate whether the i-th token is pruned or not, and $\mathcal{S}^l(x_i)$ denotes token score of the layer before the n-th stage. The token scores corresponding to different input images usually present different distributions, so using a threshold to truncate the distribution can yield a specific pruning strategy for each instance.

Considering images in the same mini-batch have different pruning configurations, for the sake of parallel training, we cannot simply discard uninformative tokens during the training process. $M^n(x_i)$ can be used to explicitly cut the connection between uninformative tokens and useful tokens. There are two masking methods, attention mask [Rao *et al.*, 2021] and activation mask [Kim *et al.*, 2021], the former acting on the attention matrix and the latter on the output of the FFN.

However, it is not easy to optimize the learnable thresholds. The pruning mask comes from comparison, blocking the gradient back propagation, making the threshold untrained. We transform the hard mask into a soft differentiable mask using the sigmoid function:

$$\tilde{M}^n(x_i) = \mathrm{Sigmoid}(T \cdot (\mathcal{S}^l(x_i) - \theta^n)), \quad n = 1, 2, 3. \tag{9}$$

To approximate the hard mask, we employ a temperature parameter T, where the Sigmoid function behaves close to the step function at a sufficiently high temperature. The soft mask is differentiable, and by using the gradient straight through estimator (**STE**), we are able to optimize the learnable threshold as normal.

3.4 Budget-Aware Training

We achieve token pruning by constraining the target computation across the dataset. Compared to DynamicViT [Rao *et al.*, 2021], which manually sets the token sparsity ratio at each stage to indirectly control the complexity, our method automatically achieves a good trade-off between accuracy and complexity given only a budget. Specifically, we propose a budget-aware loss ($\mathcal{L}_{\text{FLOPs}}$). Given a mini-batch inputs x with size of B, we can obtain their average FLOPs and then make the actual computational cost close to the target budget using MAE loss as Equation (10):

$$\mathcal{L}_{\text{FLOPs}} = \left\| \mathrm{FLOPs}(x, \Theta)/B - t \right\|_1, \tag{10}$$

where FLOPs is a function to calculate the actual operations for different inputs under the effect of all thresholds Θ, and t is the expected complexity. The intention of designing the budget-aware loss is consistent with our sample-adaptive approach. The learnable thresholds are optimized to the appropriate range in a data-driven manner under the budget constraint, without imposing other artificial restrictions.

Our training objective include other two parts. The first part is the regular cross-entropy loss ($\mathcal{L}_{\text{CE}}$) as following equation:

$$\mathcal{L}_{\text{CE}} = \mathrm{CrossEntropy}(y, \hat{y}) \tag{11}$$

where y denotes the ground truth labels and $\hat{y}$ the Softmax output. To further improve the performance, we consider transferring the knowledge of the full model to the compressed network during training. Let the distribution $\hat{y}_t$ denote the prediction of the teacher network, and we use KL divergence to minimize the gap between the output of the student network and the teacher network. For vision transformers like LV-ViT [Jiang *et al.*, 2021] with an additional linear layer to integrate all image tokens, it also needs to be aligned with the teacher network. Therefore, distillation loss ($\mathcal{L}_{\text{distill}}$) can be written as:

$$\mathcal{L}_{\text{distill}} = \mathrm{KL}(\hat{y}, \hat{y}_t) \quad \text{or} \quad \mathcal{L}_{\text{distill}} = \mathrm{KL}(\hat{y}, \hat{y}_t) + \mathrm{KL}(\hat{z}, \hat{z}_t). \tag{12}$$

The overall training objective is a combination of the above three components:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_1 \mathcal{L}_{\text{FLOPs}} + \lambda_2 \mathcal{L}_{\text{distill}}, \tag{13}$$

where λ is used to control the loss balance, and we set $\lambda_1 = 2$, $\lambda_2 = 0.5$ in our experiments.

Inference. Learnable thresholds are fixed after the training. Given an input image to do inference, we only need

Model	Params (M)	GFLOPs	Top-1 Acc (%)	Throughput (img/s)	Latency (ms)
DeiT-S [Touvron *et al.*, 2021a]	22.1	4.6	79.8	770	6.16
DyViT/ρ=0.7 [Rao *et al.*, 2021]	22.8	2.9	79.3 (-0.5)	1155 (+50%)	7.95 (+29%)
PS-ViT [Tang *et al.*, 2021]	22.1	2.6	79.4 (-0.4)	-	-
IA-RED^2 [Pan *et al.*, 2021]	-	3.2	79.1 (-0.7)	-	-
Evo-ViT [Xu *et al.*, 2022]	22.1	3.0	79.4 (-0.4)	1143 (+48%)	8.66 (+41%)
EViT-DeiT-S/ρ=0.7 [Liang *et al.*, 2022]	22.1	3.0	79.5 (-0.3)	1149 (+49%)	7.3 (+19%)
A-ViT [Yin *et al.*, 2021]	22.1	3.6	78.6 (-1.2)	-	-
AS-DeiT-S/f=0.65 (Ours)	22.1	3.0	79.6 (-0.2)	1192 (+55%)	6.56 (+6%)
DyViT/ρ=0.5 [Rao *et al.*, 2021]	22.8	2.2	77.5 (-2.3)	-	-
EViT-DeiT-S/ρ=0.5 [Liang *et al.*, 2022]	22.1	2.3	78.5 (-1.3)	1494 (+94%)	7.19 (+17%)
AS-DeiT-S/f=0.5 (Ours)	22.1	2.3	78.7 (-1.1)	1520 (+97%)	6.38 (+4%)
LV-ViT-S[Jiang *et al.*, 2021]	26.2	6.6	83.3	571	9.24
DyViT-LV-S/ρ=0.7 [Rao *et al.*, 2021]	26.9	4.6	83.0 (-0.3)	807 (+41%)	11.06 (+20%)
EViT-LV-S/ρ=0.7 [Liang *et al.*, 2022]	26.2	4.7	83.0 (-0.3)	-	-
AS-LV-S/f=0.7 (Ours)	26.2	4.6	83.1 (-0.2)	884 (+55%)	9.20 (-0%)
DyViT-LV-S/ρ=0.5 [Rao *et al.*, 2021]	26.9	3.7	82.0 (-1.3)	1011 (+77%)	11.04 (+19%)
EViT-LV-S/ρ=0.5 [Liang *et al.*, 2022]	26.2	3.9	82.5 (-0.8)	-	-
AS-LV-S/f=0.6 (Ours)	26.2	3.9	82.6 (-0.7)	1023 (+80%)	9.31 (+1%)
LV-ViT-M[Jiang *et al.*, 2021]	55.8	12.7	84.1	317	11.63
DyViT-LV-M/ρ=0.8 [Rao *et al.*, 2021]	57.1	9.6	83.9 (-0.2)	-	-
AS-LV-M/f=0.76 (Ours)	55.8	9.6	84.0 (-0.1)	657 (+107%)	11.43 (-2%)
DyViT-LV-M/ρ=0.7 [Rao *et al.*, 2021]	57.1	8.5	83.8 (-0.3)	476 (+50%)	13.29 (+14%)
AS-LV-M/f=0.67 (Ours)	55.8	8.5	83.9 (-0.2)	801 (+153%)	11.53 (-1%)

Table 1: Comparisons with the previous token pruning methods. Use ρ to denote the token keeping rate and f to indicate the proportion of budget. GFLOPs represents the average computational costs across the dataset. The throughput metric is measured on a single NVIDIA 2080Ti GPU using a fixed batch size 64 and hardware latency is the average elapsed time of 100 inferences with a single image on the same machine.

the intermediate result of MHSA to get the token score effortlessly. The pruning process is also simple enough that our method only needs one comparison to know which tokens are to be kept, saving the topK computation cost compared to the previous work [Rao *et al.*, 2021; Xu *et al.*, 2022; Liang *et al.*, 2022].

4 Experiments

4.1 Implementation Details

Our experiments are conducted on the ImageNet-1K [Deng *et al.*, 2009] classification dataset, with token pruning performed on the popular DeiT [Touvron *et al.*, 2021a] and LV-ViT [Jiang *et al.*, 2021] models. We finetune the pre-trained model by 30 epoch to obtain the compressed network, and most of the training settings stay the same as the originals.

4.2 Main Results

Performance Comparisons With Existing Pruning Methods. We test the Top-1 accuracy, throughput and hardware latency of three models, DeiT-S, LVViT-S and LVViT-M, under different pruning budgets and compare them with previous token pruning work. The experimental results on accuracy are illustrated in Table 1. Compared to previous work, our proposed adaptive sparse ViT achieves state-of-the-art performance with similar complexities. For all models, the Top-1 accuracy degradation of our pruned models is controlled within 0.2% when the computation decreases by 30%∼35%. When the compression rate of DeiT-small is further increased to 50%, the advantage of sample-adaptive token sparsity becomes more obvious over the fixed pruning rate approaches [Rao *et al.*, 2021; Liang *et al.*, 2022], as they may be forced to discard some important tokens. Moreover, our method is far better than the sample-adaptive [Pan *et al.*, 2021; Yin *et al.*, 2021], owing to the more accurate token scoring mechanism. The reinforcement learning strategy employed by IA-RED^2 [Pan *et al.*, 2021] is difficult to converge.

Model	ASM	HS	Acc (%)	Throughput (img/s)
			79.36	1099
		✓	79.36	1086
	✓		79.56 (+0.2)	1198
AS-DeiT-S	✓	✓	79.63 (+0.27)	1192

Table 2: Effectiveness of each module.

Method	Acc (%)	GFLOPs
+ attention_mask	79.63	3.0
+ activation_mask	78.04 (-1.59)	3.0
w/o $\mathcal{L}_{\text{distill}}$	79.46 (-0.17)	3.0

Table 3: Effect of different masking strategies and distillation.

	Method	ρ=0.9	ρ=0.8	ρ=0.7	ρ=0.5
pretrained	vanilla	79.77	79.24	78.51	73.72
pretrained	HS	79.79	79.29	78.51	73.78
finetuned	AS-ViT	79.8	79.7	79.6	78.7

Table 4: Comparison with the raw attention of the pre-trained model.

Efficiency Comparisons With Existing Pruning Methods. The experimental results on efficiency are in Table 1. Note that some previous methods did not release code and trained weights, so we cannot compare and report them. Compared to previous work, AS-ViT achieves the best results in terms of accuracy, throughput and hardware latency. In contrast to methods like DynamicViT [Rao *et al.*, 2021] that use extra modules to calculate token scores, AS-ViT relies only on the intermediate results of MHSA to evaluate tokens, thus saving expensive computations and achieving higher throughput metrics. The threshold-based comparator employed by Adaptive Sparsity Module saves the computational cost of ranking all tokens, which makes our latency metrics significantly better than previous work.

Comparisons With Other Models. We compare the complexity and accuracy of our adaptive sparse LV-ViT (abbreviated as AS-LV-ViT) with other sota models on ImageNet in Figure 1. Our AS-LV-ViT shows quite competitive performance under different complexities, with far higher throughput than other CNN [Tan and Le, 2019; Liu *et al.*, 2022b] and ViT [Chu *et al.*, 2021; Wang *et al.*, 2021; Xu *et al.*, 2021; Wu *et al.*, 2021; Touvron *et al.*, 2021b; Jiang *et al.*, 2021; Yuan *et al.*, 2021] while still providing advanced accuracy. In addition, by simply adjusting the budget, our approach achieves a better accuracy-complexity trade-off compared automatically, avoiding the tedium of manual design.

4.3 Ablation Analysis

Effectiveness of Each Sub-Module. In Table 2, we study the effect of each module in detail. **ASM** represents the Adaptive Sparsity Module, and **HS** is attention head importance weighted class attention score. In the ablation experiments, we use fixed ratio topK module instead of ASM and vanilla class attention score to replace HS. It is obvious that the Adaptive Sparsity module significantly improves the model performance compared with the fixed ratio module by 0.2%, which fully illustrates the necessity and effectiveness of sample adaptive token pruning. With ASM, the attention head weighted class attention score improves the precision by 0.07% compared to the original metric without a noticeable reduction in throughput. In addition, we can observe that HS needs to be used with ASM to have better results.

Batch Size	1	32	64	128
Acc (%)	79.63	79.60	79.58	79.61
GFLOPs	3.0	3.0	3.0	3.0

Table 5: Accuracy on ImageNet with different batch_size.

Effectiveness of Training Techniques. We apply masking strategy to achieve parallel training and knowledge distillation to stabilize the training process. The respective experimental results are listed in Table 3. We conduct experiments with DeiT-Small. Obviously, attention_mask is far better than activation_mask, which can shield the uninformative tokens from interacting with other tokens. And by transferring the knowledge of full model to the compressed model, we can further improve the accuracy.

Comparison With Raw Attention Baseline of Pre-Trained Model. In this part, we consider the attention of the pre-trained model as a token evaluation metric and discard unimportant tokens in a fixed proportion. In addition, we apply head importance weighted class attention score (**HS**) directly on the raw attention baseline to fully demonstrate the effectiveness of our method. The data in Table 4 present the accuracy of each method at different keeping rates. The attention map of the pre-trained model is no longer reliable and the accuracy decreases significantly as the compression ratio gradually increases. While our method can still achieve the similar performance of the original model by fine-tuning. Furthermore, our head importance weighted token score can produce positive results without training.

Batch Inference. Our approach achieves both sample-adaptive and batch-adaptive token pruning. The fact that AS-ViT is well suited for single-image computation scenarios does not mean that it cannot be used for parallel inference. Since we use the average complexity of mini-batch to calculate the budget-aware loss, we can also do parallel inference from the mean value of the number of kept tokens within a batch. And the experimental results in the Table 5 show that this does not cause significant accuracy degradation.

Pruning Location. Referring to previous works [Rao *et al.*, 2021; Liang *et al.*, 2022], we adopts a progressive token sparse strategy. The position and number of our Adaptive Sparsity Module are need to consider. We experiment with several insertion configurations to demonstrate that the current strategy is accuracy-latency optimal. The results are given in Table 6. The current method has higher accuracy compared to the [3,6,9] pruning strategy. This may be attributed to the early layer's class attention score not being stable enough, causing some information tokens to be discarded incorrectly. When increasing to 4 modules, there is no significant change in throughput and latency. We further insert thresholds from the 3rd to 11th layers, and the training becomes unstable and brings severe accuracy degradation, which we speculate is attributed to the difficulty of optimizing multiple thresholds in the limited fine-tuning. In addition, frequent reorganization of tokens causes non-negligible time consumption.

Location	GFLOPs	Top-1 Acc (%)	Throughput (img/s)	Latency (ms)
$[\mathbf{4}, \mathbf{7}, \mathbf{10}]$	3.0	**79.63**	1192	6.56
$[3,\ 6,\ 9]$	3.0	79.46	1184	6.54
$[3,\ 5, 7,\ 9]$	3.0	79.46	1191	6.60
$[4, 6, 8, 10]$	3.0	79.5	1195	6.65
$[3, 4, ..., 10, 11]$	3.0	78.8	1146	7.6

Table 6: Impacts of module insertion configurations on accuracy and speed. We use the DeiT-small model with 65% FLOPs as the baseline.

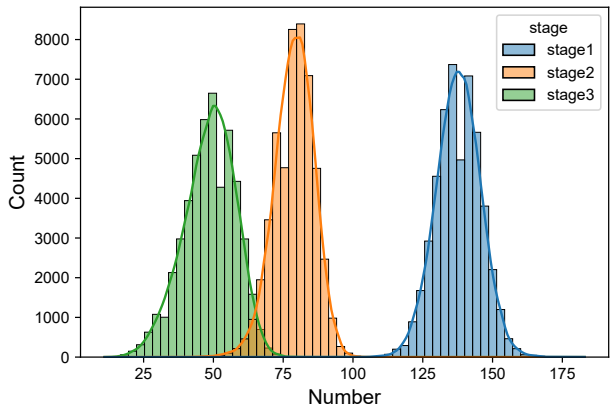

Figure 5: Distribution of the token number at different stages.

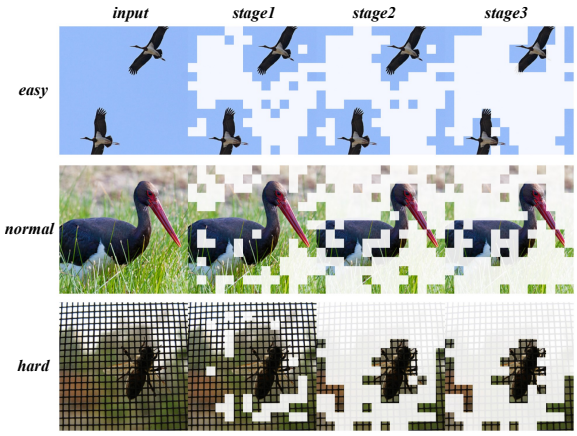

Figure 6: Visualization results of token pruning for samples with different recognition difficulties.

Method	Resolution	Acc (%)	GFLOPs
DeiT-B [Touvron *et al.*, 2021a]	224	81.8	17.5
DyViT-B [Rao *et al.*, 2021]	224	81.3 (-0.5)	11.2
EViT-DeiT-B [Liang *et al.*, 2022]	224	81.3 (-0.5)	11.5
IA-RED2 [Pan *et al.*, 2021]	224	80.3 (-1.5)	11.8
AS-DeiT-B	224	81.4 (-0.4)	11.2
DeiT-B [Touvron *et al.*, 2021a]	384	82.9	49.4
IA-RED2 [Pan *et al.*, 2021]	384	81.9 (-1.0)	34.7
AS-DeiT-B	384	82.7 (-0.2)	34.6

Table 7: Experimental results of token pruning in large model and at different input resolutions.

Performance on Large Model and Different Resolutions. To fully demonstrate the effectiveness of our method, we perform token sparse on the large baseline model DeiT-Base at different input resolutions. As illustrated in the Table 7, AS-ViT performs better than previous work under the same resolution and complexity. When using larger resolutions, our method is significantly better than IA-RED^2 [Pan *et al.*, 2021]. Furthermore, at a resolution of 384x384, the accuracy degradation is smaller compared to 224x224, suggesting that there is greater redundancy at higher resolutions, which can improve efficiency through token pruning.

Visualization. To analyze the pruning behavior of our adaptive token sparse method on different images, we count the amount of kept tokens of the AS-DeiT-S model in each stage for ImageNet validation dataset images and plot their distribution in Figure 5. The number of each stage basically shows a Gaussian distribution, peaking at a certain value. This suggests that a fixed proportion of pruning is also relatively reasonable, since the easy and difficult samples are only a minority. However, our adaptive token sparse method can give the corresponding pruning configurations for samples with different recognition difficulties, which is reflected in the distribution extending to both sides. Further, we select three representative images for visualization of token pruning results in Figure 6. For easy samples, AS-ViT discards plenty of useless tokens in the early stage to save computational cost, While for complicated and hard to recognize instances, the model tends to discard more tokens in later stages with higher confidence.

5 Conclusion

In this work, we propose a sample-adaptive token pruning method, which effortlessly evaluates token importance via fully exploiting the MHSA mechanism, and then introduces learnable thresholds to accomplish pruning. Our proposed budget-aware loss can effectively constrain the thresholds so that the pruned model reaches the complexity budget. Compared to previous work, our approach does not require additional sub-networks to compute token scores and also performs specific pruning strategies for different samples just by comparing with thresholds. Experimental results on various models show that our AS-ViT greatly improves the throughput and achieves lower latency without significantly affecting the accuracy. In the future, we will transfer token pruning to downstream tasks and combining it with static parameters pruning for further acceleration.

Contribution Statement

Xiangcheng Liu and Tianyi Wu contribute equally to this work. This work was done by Xiangcheng Liu as an intern at the Institute of Deep Learning, Baidu Research. The corresponding author is Guodong Guo.

References

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020.

[Chen *et al.*, 2021] 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.

[Chu *et al.*, 2021] 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.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[Dosovitskiy *et al.*, 2020] 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 *International Conference on Learning Representations (ICLR)*, 2020.

[Frankle and Carbin, 2018] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018.

[Goyal *et al.*, 2020] Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. Power-bert: Accelerating bert inference via progressive word-vector elimination. In *International Conference on Machine Learning*, pages 3690–3699. PMLR, 2020.

[Han *et al.*, 2015] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[He *et al.*, 2018] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. *arXiv preprint arXiv:1808.06866*, 2018.

[Jiang *et al.*, 2021] Zihang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *arXiv preprint arXiv:2104.10858*, 2021.

[Kim *et al.*, 2021] Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. Learned token pruning for transformers. *arXiv preprint arXiv:2107.00910*, 2021.

[Li *et al.*, 2016] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[Liang *et al.*, 2022] Youwei Liang, Chongjian GE, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Evit: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations (ICLR)*, 2022.

[Liu *et al.*, 2017] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *ICCV*, 2017.

[Liu *et al.*, 2021] 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 (ICCV)*, pages 10012–10022, 2021.

[Liu *et al.*, 2022a] Kai Liu, Tianyi Wu, Cong Liu, and Guodong Guo. Dynamic group transformer: A general vision transformer backbone with dynamic group attention. *arXiv preprint arXiv:2203.03937*, 2022.

[Liu *et al.*, 2022b] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[Luo *et al.*, 2017] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pages 5058–5066, 2017.

[Michel *et al.*, 2019] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

[Molchanov *et al.*, 2016] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

[Pan *et al.*, 2021] Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. Ia-red^2: Interpretability-aware redundancy reduction for vision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[Rao *et al.*, 2021] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit:

Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[Tan and Le, 2019] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114. PMLR, 2019.

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

[Touvron *et al.*, 2021a] 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 (ICML)*, pages 10347–10357, 2021.

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

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[Wang *et al.*, 2021] 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 (ICCV)*, pages 568–578, 2021.

[Wu *et al.*, 2021] 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.

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

[Xu *et al.*, 2021] 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*, pages 9981–9990, 2021.

[Xu *et al.*, 2022] 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 (AAAI)*, 2022.

[Yang *et al.*, 2021] Huanrui Yang, Hongxu Yin, Pavlo Molchanov, Hai Li, and Jan Kautz. Nvit: Vision transformer compression and parameter redistribution. *arXiv preprint arXiv:2110.04869*, 2021.

[Yin *et al.*, 2021] Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. Adavit: Adaptive tokens for efficient vision transformer. *arXiv preprint arXiv:2112.07658*, 2021.

[Yu *et al.*, 2022] Shixing Yu, Tianlong Chen, Jiayi Shen, Huan Yuan, Jianchao Tan, Sen Yang, Ji Liu, and Zhangyang Wang. Unified visual transformer compression. *arXiv preprint arXiv:2203.08243*, 2022.

[Yuan *et al.*, 2021] 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.

[Zheng *et al.*, 2021] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021.

[Zhu *et al.*, 2020] 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.

[Zhu *et al.*, 2021] Mingjian Zhu, Yehui Tang, and Kai Han. Vision transformer pruning. *arXiv preprint arXiv:2104.08500*, 2021.