

# Dynamic Token Pruning in Plain Vision Transformers for Semantic Segmentation

Quan Tang<sup>1\*</sup> Bowen Zhang<sup>2\*</sup> Jiajun Liu<sup>3</sup> Fagui Liu<sup>1†</sup> Yifan Liu<sup>2†</sup>  
<sup>1</sup>South China University of Technology <sup>2</sup>The University of Adelaide <sup>3</sup>CSIRO

## Abstract

Vision transformers have achieved leading performance on various visual tasks yet still suffer from high computational complexity. The situation deteriorates in dense prediction tasks like semantic segmentation, as high-resolution inputs and outputs usually imply more tokens involved in computations. Directly removing the less attentive tokens has been discussed for the image classification task but can not be extended to semantic segmentation since a dense prediction is required for every patch. To this end, this work introduces a Dynamic Token Pruning (DToP) method based on the early exit of tokens for semantic segmentation. Motivated by the coarse-to-fine segmentation process by humans, we naturally split the widely adopted auxiliary-loss-based network architecture into several stages, where each auxiliary block grades every token's difficulty level. We can finalize the prediction of easy tokens in advance without completing the entire forward pass. Moreover, we keep  $k$  highest confidence tokens for each semantic category to uphold the representative context information. Thus, computational complexity will change with the difficulty of the input, akin to the way humans do segmentation. Experiments suggest that the proposed DToP architecture reduces on average 20% ~ 35% of computational cost for current semantic segmentation methods based on plain vision transformers without accuracy degradation. The code is available through the following link: <https://github.com/zbwxp/Dynamic-Token-Pruning>.

## 1. Introduction

The Transformer [21] is a remarkable invention because of its exceptional capability to model long-range dependencies in natural language processing. It has been extended to computer vision applications and is known as the Vision Transformer (ViT) [7], by treating every image

\* Equal contribution. The work was done during Quan's visit to the Australian Institution of Machine Learning (AIML). † Corresponding author: fglu@scut.edu.cn, yifan.liu04@adelaide.edu.au.

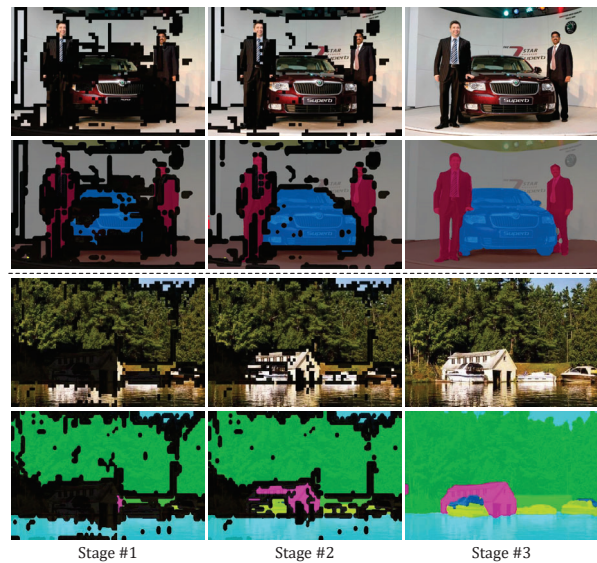


Figure 1. **Illustration of token difficulty levels by three stages using ADE20K dataset.** The network is naturally split into stages using inherent auxiliary blocks. Each sextuplet presents the early-exited/pruned tokens and their corresponding predictions successively for an image, where bright areas represent early-exited easy tokens at the current stage, while the dark ones are the kept hard tokens for the following computing.

patch as a token. Benefiting from the global multi-head self-attention, competitive results have been achieved on various vision tasks, *e.g.* image classification [7, 27], object detection [3, 32] and semantic segmentation [5, 6, 28]. However, heavy computational overhead still impedes its broad application, especially in resource-constrained environments. In semantic segmentation, the situation deteriorates since high-resolution images generate numerous input tokens. Therefore, reducing computational costs for ViT has attracted much research attention.

Since the computational complexity of vision transformers is quadratic to the token number, decreasing its magnitude is a direct path to lessen the burden of computation. There has been a line of works studying persuasive tech-

niques of token pruning regarding the image classification task. For example, DynamicViT [19] determines kept tokens using predicted probability by extra subnetworks, and EViT [13] reorganizes inattentive tokens by computing their relevance with the  $[cls]$  token. Nevertheless, *removing tokens, even if they are inattentive, can not be directly extended to semantic segmentation since a dense prediction is required for every image patch*. Most recently, Liang *et al.* [12] proposed a token reconstruction layer that rebuilds clustered tokens to address the issue.

In this work, a fresh angle is taken and breaks out of the cycle of token clustering or reconstruction. Motivated by humans' coarse-to-fine and easy-to-hard segmentation process, we progressively grade tokens by their difficulty levels at each stage. Hence for easy tokens, their predictions can be finalized in very early layers and their forward propagation can be halted early on. Consequently, only hard tokens are processed in the following layers. We refer to the process as the *early exit* of tokens. Figure 1 gives an illustration. The main body of the relatively larger objects in the image is first recognized and their process is ceased, while deeper layers progressively handle those challenging and confusing boundary regions and smaller objects. These predictions from the staged, early-exiting process can be used together with those from the completed inference. Since both outputs then form the final results jointly, it requires no token reconstruction operation and results in a simple yet effective form of efficient ViT for segmentation.

This work introduces a novel Dynamic Token Pruning (DToP) paradigm in plain vision transformers for semantic segmentation. Given that auxiliary losses [29, 30] are widely adopted, DToP divides a transformer into stages using the inherent auxiliary blocks without introducing extra modules and calculations. *While previous works discard auxiliary predictions irrespectively, we make good use of them to grade all tokens' difficulty levels*. The intuition of such a design lies in the dissimilar recognition difficulties of image patches represented by individual tokens. Easy tokens are halted and pruned early on in the ViT, while hard ones are kept to be computed in the following layers. We note that having this observation and shifting from auxiliary-loss-based architecture to DToP for token reduction is a non-trivial contribution. A possible situation exists where objects consisting of only extremely easy tokens, *e.g.* sky. As a result, DToP completely discards tokens from easy-to-recognize categories in early layers, and this causes a severe loss of contextual information for the few remaining tokens in their computations. To fully utilize the inter-class feature dependencies and uphold representative context information, we keep  $k$  highest confidence tokens for each semantic category during each pruning process. Contributions are summarized as follows:

- We introduce a dynamic token pruning paradigm based

on the early exit of easy-to-recognize tokens for semantic segmentation transformers. The finalized easy tokens at intermediate layers are pruned from the rest of the computation, and others are kept for continued processing.

- We uphold the context information by retaining  $k$  highest confidence tokens for each semantic category for the following computation, which improves segmentation accuracy by guaranteeing that enough contextual information is available even in extremely easy cases.
- We apply DToP to mainstream semantic segmentation transformers and conduct extensive experiments on three challenging benchmarks. Results suggest that DToP can reduce up to 35% computation costs without a notable accuracy drop.

## 2. Related Work

### 2.1. Semantic Segmentation Transformers

Semantic segmentation assigns each pixel a semantic category for the purpose of pixel-level image parsing. In the last decade, deep learning techniques have considerably facilitated the development of semantic segmentation approaches. Vision transformers [7, 28] now take up the baton to continue advancing the field after the great success of convolutional neural networks [8, 15]. ViT [7] adapts the standard Transformer [21] architecture to computer vision with the fewest possible modifications, which obtains competitive results and inspires recent approaches. SETR [30] first employs ViT as an encoder and incorporates a convolutional decoder, achieving impressive performance on semantic segmentation benchmarks. SegFormer [23] goes beyond the plain architecture and introduces pyramid features to acquire multi-scale contexts. Segmenter [20] uses learnable class tokens as well as the output of the encoder to predict segmentation masks, which is data-dependent. SegViT [28] further explores the capacity of the critical self-attention mechanism and proposes a novel attention-to-mask module to dynamically generate precise segmentation masks. In semantic segmentation, high-resolution images usually imply a large number of tokens. The attendant high computational complexity in vision transformers may be blamed for their limited applications.

### 2.2. Token Reduction

Since the computational complexity of vision transformers is quadratic to the length of input sequences, decreasing the number of tokens seems straightforward to reduce computation costs. DynamicViT [19] observes that an accurate image classification can be obtained by a subset of most informative tokens and proposes a dynamic token specification framework. EViT [13] demonstrates that not all to-

kens are attentive in multi-head self-attention and reorganizes them based on the attentiveness score with the  $[cls]$  token. A-ViT [24] computes a halting score for each token using the original network parameter and reserves computing for only discriminative tokens.

These token reduction approaches are carefully designed for image classification based on the intuition that removing uninformative tokens (*e.g.* backgrounds) yields a minor negative impact on the final recognition. However, things changed in semantic segmentation as we are supposed to make predictions on all image patches. Liang *et al.* [12] develop token clustering/reconstruction layers to decrease the number of tokens at middle layers and increase the number before the final prediction. Lu *et al.* [17] introduced an auxiliary PolicyNet before transformer layers to guide the token merging operation in regions with similar content. SparseViT [4] introduced a pruning routine on Swin Transformer [14] for dense prediction tasks. Differently, we perform token reduction by finalizing the prediction of easy tokens at intermediate layers and reserving computing only for hard tokens in a dynamic manner.

### 2.3. Comparisons to Prior Works

In image classification, DVT [22] determines the patch embedding granularity and generates different token numbers based on varying recognition difficulties at the image level. Easy images can be accurately predicted with a mere number of tokens, and hard ones need a finer representation. Going one step further, we base DToP on the assumption that image patches with varying contents represented by tokens are of dissimilar recognition difficulties in semantic segmentation. We can halt easy tokens and reserve only hard tokens for subsequent computing by making early predictions via auxiliary blocks at intermediate layers. As we directly combine the early predictions for easy tokens to form the final recognition results, DToP yields no information loss during token reduction and thus requires no token reconstruction operation, compared with Liang *et al.* [12].

DToP is also inspired by deep layer cascade (LC) [11] but possesses the following unique characteristics. Firstly, DToP applies to plain vision transformers and LC pyramid convolutional neural networks. The appealing architectural properties of vision transformers enable DToP to reduce computation costs without modifying the network architecture or operators, while LC requires specific region convolution. Secondly, DToP keeps  $k$  highest confidence tokens for each semantic category for the subsequent computing, which prevents easy category from halting early, contributing to the effective exploitation of contextual information.

## 3. Method

This work introduces a Dynamic Token Pruning method based on the early exit of tokens, which expedites plain vi-

sion transformers for semantic segmentation. We detail the paradigm in this section.

### 3.1. Preliminary

A conventional vision transformer [7] splits an image  $X \in \mathbb{R}^{3 \times H \times W}$  into different patches. We then obtain a sequence of  $\frac{HW}{P^2} \times C$  via patch embedding.  $H$  and  $W$  represent the image resolution,  $P$  is the patch size and  $C$  is the feature dimension. Let  $N = \frac{HW}{P^2}$  be the length of the input sequence, *i.e.* the number of tokens. Vision transformers are position-agnostic, and we generally add positional encoding to represent the spatial information of each token. The resulting sequence is denoted as  $Z_0 \in \mathbb{R}^{N \times C}$ , which serves as the input.

Vision transformers are usually developed from repeated units that contain a multi-head self-attention (MHSA) module and a feed-forward network (FFN). Layer normalization (LN) [1] and residual connection [8] are employed within such units. We refer to a unit as one layer indexed by  $l \in \{1, 2, \dots, L\}$ , and the output of each layer is marked as  $Z_l$ .

$$\begin{aligned} Z'_l &= \text{MHSA}(\text{LN}(Z_{l-1})) + Z_{l-1}, \\ Z_l &= \text{FFN}(\text{LN}(Z'_l)) + Z'_l. \end{aligned} \quad (1)$$

Note that FFN includes a non-linear activation function, *e.g.* GeLU [9].

### 3.2. Dynamic Token Pruning

Since a token is a natural representation of an image patch, we can finalize the prediction for easy tokens in advance without the need for complete forward computing by mimicking the segmentation process of humans. We refer to it as the *early exit* of tokens, where easy tokens are halted and pruned in the early stages while hard ones are preserved for calculation at the latter stages. By doing so, fewer tokens are processed in the following layers, significantly reducing the computation costs.

As shown in Figure 2, we divide a plain vision transformer backbone into  $M$  stages using its inherent auxiliary blocks  $\mathcal{H}_m$  ( $m \in \{1, 2, \dots, M\}$ ) at the end of each stage. Let  $P_m \in \mathbb{R}^{N \times K}$  represent the predicted results at the  $m$ -th stage, where  $K$  is the number of semantic category. Suppose that tokens have finished  $l_m$  layers of forward propagation at this point, then:

$$P_m = \mathcal{H}_m(Z_{l_m}). \quad (2)$$

$p_{m,n}$  coming from  $P_m$  is the maximum predicted probability of the  $n$ -th token. Previous works adopt  $P_m$  to calculate auxiliary losses during training and discard them irrespectively during inference. This work highlights that easy tokens can be correctly classified with high predictive confidence in these auxiliary outputs (*i.e.*  $P_m$ ). The proposed

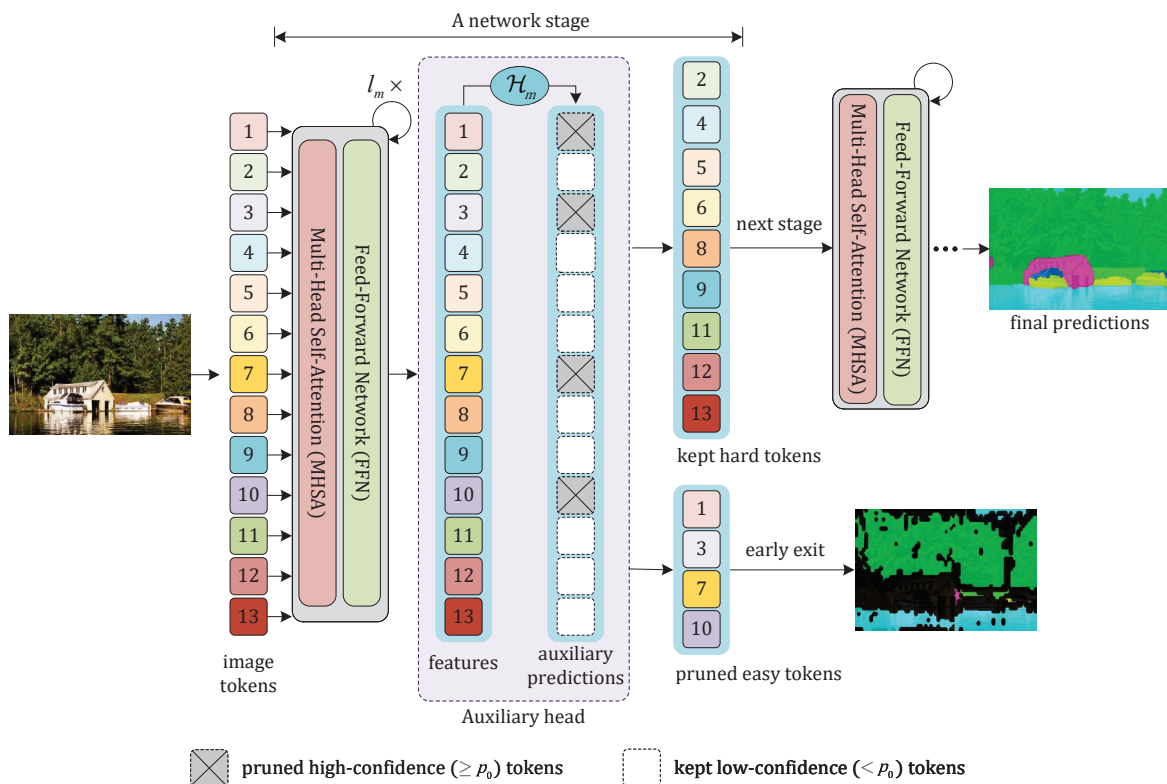


Figure 2. **Illustration of the proposed DTOP framework.** Given an existing plain vision transformer, we divide it into stages using the inherent auxiliary heads. At the final layer (indexed by  $l_m$ ) of the  $m$ -th stage, we use the auxiliary block  $\mathcal{H}_m$  to grade all token difficulty levels. We finalize the predictions of high-confidence easy tokens at the current stage and handle other low-confidence hard tokens in the following stages. The retained  $k$  highest confidence tokens for each semantic category to uphold representative context information are not presented for simplicity. Predictions from each stage jointly form the final results.

DTOP expects to fully explore their potential ability to tell apart easy and hard tokens during both training and inference.

Inspired by [10], we grade all token difficulty levels using  $P_m$  based on a simple criterion. Assume a large confidence threshold  $p_0$ , e.g. 0.9. Easy tokens are classified with higher than 90% scores, while hard ones are classified with lower scores. Since confident predictions for easy tokens are obtained, we prune them and halt their continued forward propagation. Hard tokens are reserved in computing in the following layers to achieve a reliable prediction. In other words, we prune the  $n$ -th token in  $Z_{l_m}$  if  $p_{m,n} \geq p_0$ , otherwise we keep it. After propagating an image through the whole network, we combine the predicted token labels from each stage to form the final results.

### 3.3. Query Matching Auxiliary Block

Within the DTOP framework, the auxiliary block for grading all token difficulty levels should follow two principles: capable of accurately estimating token difficulty levels and with a lightweight architecture. Therefore, we take

the most recent attention-to-mask module (ATM) [28] to achieve this goal. Specifically, a series of learnable class tokens exchange information with the encoder features using a transformer decoder. The output class tokens are used to get class probability predictions. The attention score regarding each class token is used to form a mask group. The dot product between the class probability and group masks produces the final prediction.

Two modifications are made to adapt ATM into the DTOP framework. First, we decrease the number of layers in ATM as we observe no significant performance perturbation in the DTOP framework with the original setting, which also guarantees a low computational overhead. Second, we decouple multiple cascaded ATM modules and use them as separate auxiliary segmentation heads, each with individual learnable class tokens. We note that we take the powerful ATM module to grade all token difficulty levels as an example, because a reliable estimation of tokens' segmentation difficulty will lead to a good accuracy-computation trade-off. Any other existing segmentation heads are of the same effect (see [23, 29, 30] for examples). In Section 4, we



also provide experiments with the regular FCN head [16] to validate the generality of DToP.

### 3.4. Upholding Context Information

Scenarios exist where all tokens of a specific semantic category are extremely easy to recognize, *e.g.* sky. Such tokens may be entirely removed in early layers, resulting in a loss of context information in the following layers of calculation. Practices [25, 26] indicate that fully exploiting the inter-category contextual information improves the overall semantic segmentation accuracy. To this end, we keep  $k$  highest confidence tokens for each semantic category during each pruning process. *Only the categories that appear in the current image are considered.* Given a specific semantic category, if the number of tokens with a higher than  $p_0$  score is more than  $k$ , then the top- $k$  of them are kept. Otherwise, we keep the actual number of them. These category-known tokens join in the calculation along with other low-confidence ones, so semantic information of easy category is preserved for inter-category information exchange, leading to an accurate semantic segmentation.

## 4. Experiments

### 4.1. Datasets and Metrics

**ADE20K** [31] is a widely adopted benchmark dataset for semantic segmentation. It contains about  $20k$  images for training and  $2k$  images for validation. All images are labeled with 150 semantic categories. **COCO-Stuff-10K** [2] dataset contains  $9k$  images for training and  $1k$  images for testing. Following [28], we use 171 semantic categories for experiments. **Pascal Context** [18] has a total of 10, 100 images, of which 4, 996 images are for training and 5, 104 for validation. It provides pixel-wise labeling for 59 categories, excluding the background.

Following the common convention, we use the mean intersection over union (mIoU) to evaluate the segmentation accuracy and the number of float-point operations in giga (GFLOPs) to estimate the model complexity. The computation in DToP is unevenly allocated among easy and hard samples by pruning different numbers of tokens. We thus report the average GFLOPs over the entire val/test dataset.

### 4.2. Implementation Details

We adopt the plain vision transformer incorporating the adapted ATM module as the baseline model, where ATM modules work as auxiliary heads. We follow the standard training settings in *mmsegmentation*<sup>1</sup> and use the same hyperparameters as the original paper. All reported mIoU scores are based on single-scale inputs.  $k$  is set to 5 in this work. As changing  $p_0$  within a certain range (0.90 ~ 0.98)

<sup>1</sup><https://github.com/open-mmlab/msegmentation>

Method	GFLOPs	mIoU(%)
Baseline	109.9	49.7
+ DToP@Direct (♣.0)	87.5	47.9 (-1.8)
+ DToP@Finetune (♣2.5)	86.8	<b>49.8</b> (+0.1)
+ DToP@Retrain (♣12.0)	87.5	49.1 (-0.6)

Table 1. **Comparison of training schemes.** With a short finetuning scheme, the pruned model achieves even better results than the baseline. ♣ means extra training time in hours on 8 NVIDIA A100 cards.

during training leads to similar results, we empirically fix it to 0.95 for all training processes unless specified.

### 4.3. Ablation Study

We first conduct extensive ablation studies with the ADE20K dataset [31] using ViT-Base [7] as the backbone.

#### 4.3.1 Necessity for Model Training

Using auxiliary heads for efficient training is a common convention in the semantic segmentation community, see [5, 29, 30] for examples. Generally, the auxiliary outputs are discarded at test time. As the proposed DToP grades all token difficulty levels using the auxiliary outputs, we can apply DToP to existing methods off-the-shelf during inference. Therefore, we verify the necessity for model retraining or finetuning under the proposed DToP framework. We denote DToP@Direct as directly applying DToP to the baseline model during inference. DToP@Finetune means finetuning the segmentation heads for  $40k$  iterations on the baseline model using DToP, and DToP@Retrain retraining the entire model for  $160k$  iterations.

Results are shown in Table 1. We observe that all three settings reduce the computation costs by about 20%, where DToP@Direct and DToP@Retrain lead to a significant accuracy drop while DToP@Finetune performs slightly better. Results suggest that the proposed DToP@Finetune requires only a little extra training time but significantly reduces the computational complexity while maintaining accuracy. We adopt the @Finetune setting in the following experiments. Note that the slight fluctuation in GFLOPs of the three training schemes comes from varied predictions of auxiliary heads in each individual training processes.

#### 4.3.2 Ablation for Confidence Threshold

The confidence threshold  $p_0$  is a crucial hyperparameter that decides the pruned token number in each pruning process and directly affects the trade-off between computation cost and accuracy. Quantitative results are shown in Table 2. When  $p_0 = 1$ , the model degenerates to the baseline architecture. As  $p_0$  decreases, more easy tokens are pruned as

$p_0$	0.60	0.70	0.80	0.85	0.90	0.95	1.00
<b>GFLOPs</b>	70.2	73.4	77.8	80.7	83.6	86.8	109.9
<b>mIoU(%)</b>	46.8	48.0	49.0	49.3	49.5	<b>49.8</b>	49.7

Table 2. **Ablation for confidence threshold  $p_0$ .** The results are evaluated on ADE20K with ATM head.

<b>Method</b>	$p_0$	<b>GFLOPs</b>	<b>mIoU(%)</b>
SETR	-	107.7	47.0
+ DToP@Direct	0.90	74.0	45.6 (-1.4)
+ DToP@Finetune	0.90	72.5	46.3 (-0.7)
+ DToP@Direct	0.95	78.3	46.2 (-0.8)
+ DToP@Finetune	0.95	76.5	46.8 (-0.2)
+ DToP@Direct	0.98	82.5	46.6 (-0.4)
+ DToP@Finetune	0.98	80.6	<b>47.0 (+0.0)</b>

Table 3. **Ablation results based on SETR.** About 25% of the tokens can be pruned with no performance dropped.

well as more unreliable early predictions. We observe that the performance saturates at  $p_0 = 0.95$  when using ATM as the segmentation head.

We also verify the value using SETR [30] (w/ the naive segmentation head described in FCN [16]) and show the results in Table 3. We observe that for FCN head  $p_0 = 0.98$  may be a better choice. In practice, the value can be chosen empirically with a small validation set. We also note that for SETR, DToP@Direct has already obtained a promising mIoU score of 46.6% that is only 0.4% lower than the baseline but with significantly reduced computation ( $\sim 23.4\%$ ). Some qualitative examples of how the threshold  $p_0$  affects the pruned token number and segmentation accuracy are shown in Figure 3<sup>2</sup>.

### 4.3.3 Exploration on Pruning Position

The critical insight of DToP is to finalize the prediction of easy tokens in intermediate layers and prune them in the following calculation by grading all tokens’ difficulty levels. Thus the position of auxiliary heads matters. It affects the recognition accuracy of pruned easy tokens and the trade-off between computation cost and segmentation accuracy. We conduct explorations on the pruning position  $l_m$  and show the results in Table 4. Results demonstrate that dividing the backbone into three stages with token pruning at the 6<sup>th</sup> and 8<sup>th</sup> layers achieves an expected trade-off between computation cost and segmentation accuracy. We adopt this

<sup>2</sup>Note that some pruned tokens change their final segmentation due to the attention-to-mask mechanism in ATM but will remain the same in regular FCN heads.

<b>Stages</b>	<b>Position</b>	<b>GFLOPs</b>	<b>mIoU (%)</b>
1	Baseline	109.9	49.7
2	{6}	85.7	49.4
2	{8}	92.1	49.4
3	{6, 8}	86.8	<b>49.8</b>
4	{3, 6, 8}	74.5	48.3

Table 4. **Exploration of the pruning position.** The first column indicates the number of divided stages.

<b>Method</b>	<b>Context</b>	<b>GFLOPs</b>	<b>mIoU(%)</b>
Baseline	-	109.9	49.7
Remove	×	82.6	48.7
Top-35%	×	84.6	48.7
Average	✓	83.5	48.9
Top- $k$	✓	83.6	49.5
Avg & top- $k$	✓	83.6	<b>49.7</b>

Table 5. **Comparison of different pruning methods.** All models are trained with DToP@Finetune using  $p_0 = 0.9$ .

<b>Method</b>	<b>Decode</b>	<b>Aux</b>	<b>GFLOPs</b>	<b>mIoU (%)</b>
Baseline	ATM	ATM	109.9	49.7
+ DToP@Finetune	ATM	ATM	83.6	49.5
Baseline	FCN	FCN	107.7	47.0
+ DToP@Finetune	FCN	FCN	80.6	47.0
Baseline	FCN	ATM	107.7	49.6
+ DToP@Finetune	FCN	ATM	83.4	48.4
Baseline	ATM	FCN	109.9	47.9
+ DToP@Finetune	ATM	FCN	73.3	46.9

Table 6. **Exploration of different segmentation heads.** Results in the second part uses  $p_0 = 0.98$  and others 0.9. ‘Decode’ means the final decoder head and ‘Aux’ auxiliary head.

setting in all other experiments and note that it may not be optimal on account of limited explorations.

### 4.3.4 Ablation for Pruning Method

After grading all token difficulty levels at the current stage, the specific pruning method is flexible. We experiment with four token pruning methods. Following LC [11], we remove easy tokens directly without the consideration of halting easy category information. We also prune a fixed proportion of tokens by removing the top 35% highest confidence tokens to evenly allocate computation among images. To uphold context information, an alternative is to average all easy token values into one token for each semantic category. In contrast, this work keeps  $k$  highest confidence tokens for each appeared semantic category to uphold representative context information, marked as top- $k$ . Results are shown

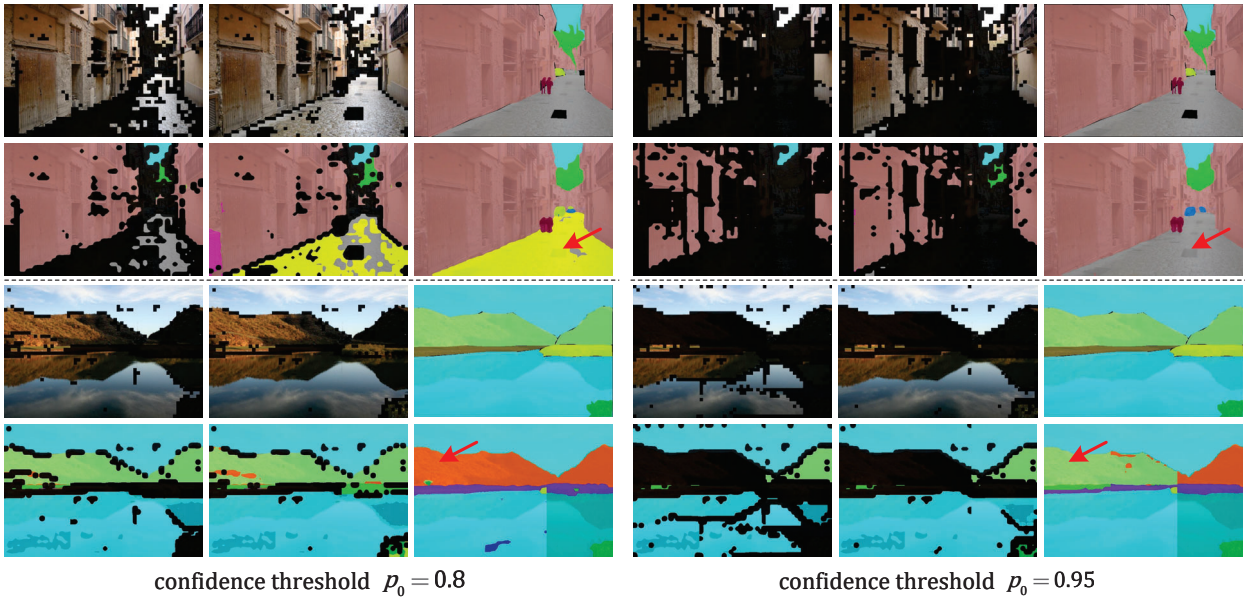


Figure 3. **Illustration of the effects for different confidence threshold.** Samples are from ADE20K dataset. For each sextuplet, we show the pruned token distribution and the ground truth (first row), as well as its corresponding segmentation results (second row). Bright areas represent pruned tokens, and those in the dark are kept tokens for the following computing. A small  $p_0$  value (left two examples) leads to more pruned tokens in early stages but results in inferior segmentation results (see the red arrow).

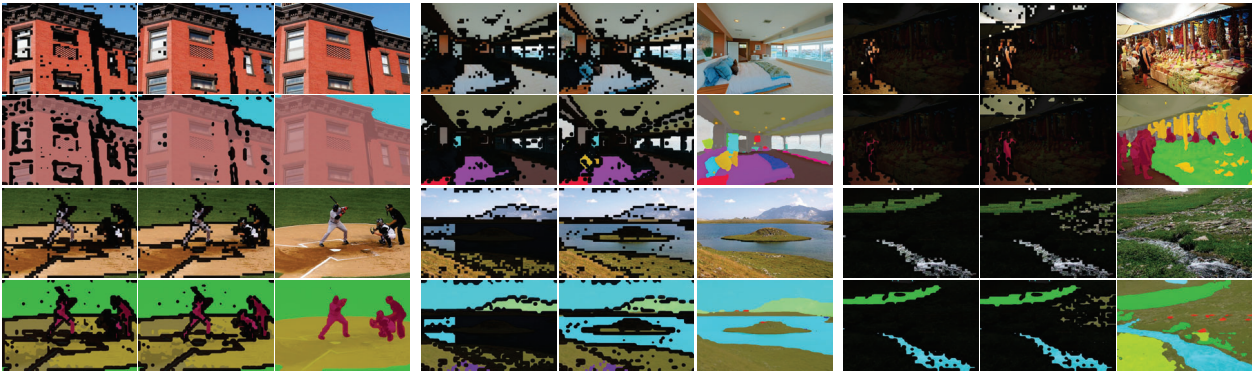


Figure 4. **Predicted results of three stages during the token pruning processes.** Examples are from ADE20K with different image complexity: most tokens are pruned (left group), the majority are pruned (middle group), and very few are pruned (right group). For each sextuplet, we show the pruned token distribution and the corresponding segmentation results at each stage.

in Table 5, where the proposed top- $k$  method outperforms others by a large margin, suggesting its effectiveness. Furthermore, we observe that combining the Average and top- $k$  strategies yields slightly better results.

#### 4.3.5 Influence of Segmentation Heads

In Table 6, we verify different segmentation heads and observe that the proposed DToP performs effectively in both ATM and FCN settings (first two parts), indicating its general applicability. To ensure a fair comparison, we select different  $p_0$  values to maintain similar GFLOPs and compare the performance. We notice that the choice of the aux-

iliary head significantly influences the performance. Particularly, the powerful ATM head provides a more accurate estimation of all tokens' difficulty levels, resulting in superior results.

#### 4.4. Application to Existing Methods

We apply the proposed DToP to two mainstream semantic segmentation frameworks in plain vision transformers. SETR [30] with DToP uses the naive upsampling decoder, and SegViT [28] adopts our adapted ATM module. Results are shown in Table 7 using three challenging benchmarks. With an appropriate confidence threshold  $p_0$ , the



Method	Backbone	$p_0$	ADE20K		Pascal Context		COCO-Stuff-10K	
			mIoU(%)	GFLOPs	mIoU(%)	GFLOPs	mIoU(%)	GFLOPs
SETR [30]	ViT-Base	-	47.0	107.7	58.1	92.4	41.2	107.7
+ DToP@Finetune	ViT-Base	0.90	46.3	72.5	57.5	61.4	40.6	77.6
+ DToP@Finetune	ViT-Base	0.98	<u>47.0</u>	80.6	<b>58.2</b>	69.1	<u>40.9</u>	86.4
SegViT [28]	ViT-Large	-	53.3	617.0	63.0	315.4	47.4	366.9
+ DToP@Finetune	ViT-Large	0.90	52.4	380.3	62.2	206.1	46.6	253.1
+ DToP@Finetune	ViT-Large	0.95	<u>52.8</u>	412.8	<u>62.7</u>	224.3	<u>47.1</u>	276.2

Table 7. **Main results on three semantic segmentation benchmarks.** We apply the proposed DToP with the finetuning training scheme to current state-of-the-art semantic segmentation networks based on plain vision transformers. GFLOPs is the average number of the whole validation dataset. We perform token pruning at  $\{8^{th}, 16^{th}\}$  layers for ViT-Large.



Figure 5. **Visualized results.** The segmentation results are predicted on ADE20K (first row), Pascal Context (middle row), and COCO-Stuff-10K (last row). The model is SegViT with DToP@Finetune based on ViT-Large.

proposed DToP can reduce on average 20% ~ 35% computation cost without notable accuracy degradation. More specifically, SETR with DToP@Finetune reduces 25.2% computation cost (FLOPs 107.7G  $\rightarrow$  80.6G) without mIoU drop on ADE20K and even obtains a slightly better mIoU (58.1%  $\rightarrow$  58.2%) on Pascal Context dataset. SegViT with DToP@Finetune based on ViT-large reduced about 35% computation with only 0.5% mIoU lower on ADE20K.

A qualitative comparison regarding the pruned token number of different images is presented in Figure 4. We see that most tokens are pruned at very early stages for images of simple scenarios. For complex scene images, most tokens remain until the final prediction. Consequently, the computation is unevenly allocated among images by adjusting the pruned token number, yielding a considerable improvement in computation efficiency. We also observe that pruned easy tokens are primarily located at the central area of objects, while kept hard tokens are located on the boundaries, similar to the segmentation process by humans. Some visualized predictions are shown in Figure 5.

## 5. Conclusion

This work studies the problem of reducing computation costs for existing semantic segmentation based on plain vision transformers. A Dynamic Token Pruning paradigm is proposed based on the early exit of tokens. Motivated by the coarse-to-fine segmentation process by humans, we assume that different tokens representing image regions have dissimilar recognition difficulties and grade all tokens' difficulty levels using the inherent auxiliary blocks. To this end, we finalize the predictions of easy tokens at intermediate layers and halt their forward propagation, which dynamically reduces computation. We further propose a strategy to uphold context information by preserving extremely easy semantic categories after token pruning. Extensive experimental results suggest that the proposed method achieves compelling performance.

Similar to all other dynamic networks, DToP can not take full advantage of the calculation efficiency of a mini-batch. We will make optimization in the future and further expedite vision transformers using the proposed DToP.



**Acknowledgements** This work was partially supported by the Guangdong Major Project of Basic and Applied Basic Research (2019B030302002), the Science and Technology Major Project of Guangzhou (202007030006), the Science and Technology Project of Guangdong Province (2021B1111600001), and the Major Key Project of PCL (PCL2023AS7-1) to F. Liu.

## References

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv: Comp. Res. Repository*, 2016. 3
- [2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocosuff: Thing and stuff classes in context. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1209–1218, 2018. 5
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. Eur. Conf. Comp. Vis.*, pages 213–229. Springer, 2020. 1
- [4] Xuanyao Chen, Zhijian Liu, Haotian Tang, Li Yi, Hang Zhao, and Song Han. Sparsevit: Revisiting activation sparsity for efficient high-resolution vision transformer. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2061–2070, 2023. 3
- [5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1290–1299, 2022. 1, 5
- [6] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Pixel classification is not all you need for semantic segmentation. *Proc. Advances in Neural Inf. Process. Syst.*, 34:17864–17875, 2021. 1
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. Int. Conf. Learn. Representations*, 2021. 1, 2, 3, 5
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 770–778, 2016. 2, 3
- [9] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv: Comp. Res. Repository*, 2016. 3
- [10] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proc. Int. Conf. Learn. Representations*, 2017. 4
- [11] Xiaoxiao Li, Ziwei Liu, Ping Luo, Chen Change Loy, and Xiaoou Tang. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3193–3202, 2017. 3, 6
- [12] 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. *Advances in Neural Information Processing Systems*, 35:35462–35477, 2022. 2, 3
- [13] Youwei Liang, Chongjian GE, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. EVit: Expediting vision transformers via token reorganizations. In *Proc. Int. Conf. Learn. Representations*, 2022. 2
- [14] 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 *Proc. IEEE Int. Conf. Comp. Vis.*, pages 10012–10022, 2021. 3
- [15] Zhuang Liu, Hanzhi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 11976–11986, 2022. 2
- [16] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3431–3440, 2015. 5, 6
- [17] Chenyang Lu, Daan de Geus, and Gijs Dubbelman. Content-aware token sharing for efficient semantic segmentation with vision transformers. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 23631–23640, 2023. 3
- [18] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 891–898, 2014. 5
- [19] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Proc. Advances in Neural Inf. Process. Syst.*, 34:13937–13949, 2021. 2
- [20] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 7262–7272, 2021. 2
- [21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proc. Advances in Neural Inf. Process. Syst.*, 30, 2017. 1, 2
- [22] Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Proc. Advances in Neural Inf. Process. Syst.*, 34:11960–11973, 2021. 3
- [23] Enze Xie, Wenhui Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Proc. Advances in Neural Inf. Process. Syst.*, 34:12077–12090, 2021. 2, 4
- [24] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 10809–10818, 2022. 3
- [25] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene seg-

- mentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 12416–12425, 2020. [5](#)
- [26] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1857–1866, 2018. [5](#)
- [27] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 558–567, 2021. [1](#)
- [28] Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, et al. Segvit: Semantic segmentation with plain vision transformers. *Advances in Neural Information Processing Systems*, 35:4971–4982, 2022. [1](#), [2](#), [4](#), [5](#), [7](#), [8](#)
- [29] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2881–2890, 2017. [2](#), [4](#), [5](#)
- [30] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 6881–6890, 2021. [2](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [31] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 633–641, 2017. [5](#)
- [32] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *Proc. Int. Conf. Learn. Representations*, 2021. [1](#)